

# Energiemanagement für drahtlose tief eingebettete Systeme

Feingranulares, faires Management zum Erreichen von Lebenszeitzielen  
bei knappen Energieressourcen

Von der Fakultät für MINT - Mathematik, Informatik, Physik,  
Elektro- und Informationstechnik  
der Brandenburgischen Technischen Universität Cottbus - Senftenberg

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften  
(Dr.-Ing.)

genehmigte Dissertation

vorgelegt von

Herrn Diplom Informatiker

André Sieber

geboren am 08.09.1980 in Berlin-Mitte

Gutachter: Prof. Dr.-Ing. Jörg Nolte

Gutachter: Prof. Dr.-Ing. Rolf Kraemer

Gutachter: Prof. Dr.-Ing. Falko Dressler

Tag der mündlichen Prüfung: 01.07.2016



# Abstract

Wireless embedded systems can be used in broad application ranges, for example in the internet of things and wireless sensor networks. They should run for years on a single set of batteries, without external interference. Energy supply and consumption play a major role, but are vulnerable to variances. To be able to reach runtime goals, e.g. maintenance intervals, dynamic energy management is necessary.

This thesis introduces, describes and evaluates such a management approach. In contrast to existing approaches, variances are addressed on all three management levels: energy supply, energy consumption and energy allowance. For the energy supply a battery observation method is presented, which enables the steering of the application without complex computations of the state-of-charge. A software-based approach determines fine-grained information about the energy consumption of the system and includes variances caused by differences of the supply voltage and the voltage converter efficiency. The energy allowance to different application parts and their management is based on energy budgets and provides different policies to react to changes in the energy demand of single application(-parts).

Embedded Systems, Wireless Sensor Networks, Energy Management, Energy Awareness, Lifetime Goals

## Zusammenfassung

Drahtlose eingebettete Systeme haben vielfältige Anwendungsmöglichkeiten, sei es im Internet der Dinge oder in Sensornetzen. Diese Systeme sollen dabei ohne Eingriff von außen eine hohe Laufzeit erreichen. Die Energieversorgung und der Verbrauch spielen dabei entscheidende Rollen, sind jedoch Varianzen unterworfen. Sollen Wartungsintervalle oder Lebenszeitziele erreicht werden, ist deshalb ein dynamisches Energiemanagement unabdingbar.

In dieser Doktorarbeit wird ein solcher Ansatz vorgestellt, umgesetzt und evaluiert. Im Gegensatz zu existierenden Ansätzen werden Varianzen auf allen drei Ebenen des Managements (Energieversorgung, Energieverbrauch und Energieverwaltung) adressiert. Für die Energieversorgung wird eine Batterieüberwachung vorgestellt, welche eine Steuerung des Verbrauchs ohne komplexe Berechnung des Ladezustands ermöglicht. Der Verbrauch des Systems wird durch einen Software-basierten Ansatz feingranular ermittelt und bezieht Varianzen durch Änderungen der Spannung und der Effizienz von Spannungswandlern ein. Die Verwaltung und Zuordnung der Energie zu Anwendungszielen erfolgt mittels dynamischer Energiebudgets und bietet verschiedene Strategien, um auf Änderungen des Bedarfs adäquat reagieren zu können.

Eingebettete Systeme, Drahtlose Sensornetze, Energiemanagement, Energiebewusstsein, Lebenszeitziele



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Beitrag der Arbeit . . . . .	5
1.2	Zugrundeliegende Veröffentlichungen . . . . .	6
1.3	Struktur der Arbeit . . . . .	7
<b>2</b>	<b>Energieversorgung und -verbrauch in drahtlosen tief eingebetteten Systemen</b>	<b>9</b>
2.1	Energieversorgung . . . . .	10
2.1.1	Primär- und Sekundärzellen . . . . .	10
2.1.2	Superkondensatoren . . . . .	17
2.1.3	Energiegewinnung aus der Umwelt . . . . .	17
2.1.4	Spannungsregler . . . . .	20
2.2	Komponenten und ihre Benutzung . . . . .	21
2.2.1	Prozessor . . . . .	21
2.2.2	Kommunikation . . . . .	27
2.2.3	Sensorik & Aktuatorik . . . . .	34
2.3	Messung versus Datenblatt . . . . .	37
2.3.1	Messaufbau . . . . .	38
2.3.2	Messergebnisse . . . . .	40
2.3.3	Mögliche Konsequenzen . . . . .	48
2.4	Zusammenfassung . . . . .	49
<b>3</b>	<b>Ladezustandsüberwachung zum Erreichen von Lebenszeitzielen</b>	<b>51</b>
3.1	Problembeschreibung . . . . .	52
3.2	Stand der Technik . . . . .	53
3.2.1	Hardware-basierte Ansätze . . . . .	53
3.2.2	Modell-basierte Ansätze . . . . .	55
3.2.3	Messbasierte Ansätze . . . . .	57
3.2.4	Zusammenfassung der Ansätze . . . . .	58
3.3	Hinweis-basiertes Entlademanagement . . . . .	59
3.3.1	Versorgungsspannungsbeobachtung . . . . .	60

3.3.2	Abstraktes Batteriemodell . . . . .	62
3.4	Evaluation . . . . .	63
3.4.1	Experiment 1 - Wirksamkeit der Versorgungsspannungsbeobachtung . . . . .	64
3.4.2	Experiment 2 - Versorgungsspannungsbeobachtung für Superkondensatoren . . . . .	66
3.4.3	Experiment 3 - abstraktes Batteriemodell . . . . .	68
3.4.4	Experiment 4 - verringerte Kapazität . . . . .	70
3.4.5	Experiment 5 - temperaturbedingte Störungen . . . . .	73
3.5	Zusammenfassung . . . . .	77
<b>4</b>	<b>Verbrauchserfassung im Feld</b>	<b>79</b>
4.1	Problembeschreibung . . . . .	80
4.2	Stand der Technik . . . . .	81
4.2.1	Hardware-basierte Ansätze . . . . .	82
4.2.2	Software-basierte Ansätze . . . . .	84
4.2.3	Hybride Ansätze . . . . .	85
4.2.4	Zusammenfassung der Ansätze . . . . .	86
4.3	Verbrauchserfassung auf Geräteebe . . . . .	87
4.3.1	Verbauchsmodell . . . . .	88
4.3.2	Erzeugung der Verbrauchsdaten . . . . .	90
4.4	Evaluation . . . . .	91
4.4.1	Mehraufwand . . . . .	91
4.4.2	Vergleich von Modell und Messung . . . . .	94
4.4.3	Adaption an die Versorgungsspannung . . . . .	98
4.4.4	Adaption an Spannungsregler . . . . .	98
4.4.5	Anwendungsbeispiel: Adaption des Anwendungsintervalls . . . . .	99
4.4.6	Anwendungsbeispiel: Verbrauchserfassung eines Sensornetzes . . . . .	100
4.5	Zusammenfassung . . . . .	103
<b>5</b>	<b>Dynamische feingranulare Energiebudgets</b>	<b>105</b>
5.1	Problembeschreibung . . . . .	106
5.2	Stand der Technik . . . . .	107
5.2.1	Direkte Steuerung . . . . .	108
5.2.2	Indirekte Steuerung . . . . .	113
5.2.3	Zusammenfassung der Ansätze . . . . .	118
5.3	Dynamische Energiebudgets . . . . .	122
5.3.1	Zuordnung der Budgets zu Systemkomponenten . . . . .	125
5.3.2	Zuordnung von asynchronem Verbrauch . . . . .	127
5.4	Strategien zur Budgetverwaltung . . . . .	128

5.4.1	Verteilung der Energie . . . . .	128
5.4.2	Ausnahmesituationen . . . . .	131
5.5	Verwendung von Budgets zur Kontrolle der Anwendung . . . . .	137
5.5.1	Bestimmung von Anwendungsintervallen . . . . .	138
5.5.2	Bestimmung von Servicestufen . . . . .	138
5.5.3	Quantisierung durch Token Buckets . . . . .	139
5.6	Evaluation . . . . .	140
5.6.1	Mehraufwand . . . . .	140
5.6.2	Verteilung der Energie . . . . .	143
5.6.3	Unzureichende Systemenergie . . . . .	145
5.6.4	Strategien für Mehr- und Minderbedarf . . . . .	146
5.6.5	Limitierung der angesparten Energie . . . . .	150
5.6.6	Langfristiger Fehlbedarf . . . . .	151
5.6.7	Zusammenfassung der Strategien . . . . .	154
5.6.8	Systemverhalten im Vergleich mit Energy Levels . . . . .	155
5.7	Zusammenfassung . . . . .	157
<b>6</b>	<b>Feldversuche</b>	<b>159</b>
6.1	Geräteüberwachung . . . . .	159
6.1.1	Aufbau . . . . .	160
6.1.2	Auswertung . . . . .	161
6.2	Hydrologisches Umweltmonitoring . . . . .	164
6.2.1	Aufbau . . . . .	166
6.2.2	Auswertung der Ausbringung . . . . .	168
6.2.3	Auswertung des Managements . . . . .	169
6.3	Zusammenfassung . . . . .	172
<b>7</b>	<b>Zusammenfassung und Ausblick</b>	<b>173</b>
7.1	Zusammenfassung . . . . .	174
7.2	Ausblick . . . . .	175
<b>A</b>	<b>Abbildungen</b>	<b>177</b>
<b>B</b>	<b>Tabellen</b>	<b>181</b>



# Abbildungsverzeichnis

2.1	Typischer Aufbau eines drahtlosen tief eingebetteten Systems . . . .	9
2.2	Auswirkung der Temperatur auf Selbstentladung von Batterien . . .	12
2.3	Auswirkung der Temperatur auf Energiedichte von Batterien . . . .	13
2.4	Einfluss der Temperatur auf die Spannung . . . . .	13
2.5	Spannungskurven von Primär- und Sekundärzellen . . . . .	14
2.6	Typische Entladekurve . . . . .	14
2.7	Einflüsse auf die Kapazität . . . . .	15
2.8	Auswirkung der Last auf Primär- und Sekundärzellen . . . . .	15
2.9	Aufbau eines Systems mit Energiegewinnung aus der Umwelt . . . .	19
2.10	Effizienz NCP1400A33T1 . . . . .	20
2.11	Duty Cycling . . . . .	21
2.12	Darstellung einer vereinfachten Sensornetzanwendung . . . . .	24
2.13	Vergleich der Aktivität verschiedener MAC-Konzepte . . . . .	30
2.14	Adaptive Datenerfassung . . . . .	36
2.15	Untersuchter Sensorknoten . . . . .	38
2.16	Schematischer Messaufbau . . . . .	40
2.17	Verteilung der gemessenen Stromaufnahme . . . . .	41
2.18	Detaillierte Verteilung . . . . .	42
2.19	Vom Median abweichender Verbrauch, Teil 1 . . . . .	44
2.20	Vom Median abweichender Verbrauch, Teil 2 . . . . .	45
2.21	Abweichung der Messergebnisse vom Datenblatt . . . . .	46
2.22	Abweichung der Verbrauchsannahme . . . . .	48
3.1	Anwendungssteuerung unter Einbeziehung der Batterie . . . . .	51
3.2	Ermittlung der in einem Zeitschritt zur Verfügung stehenden Energie	52
3.3	Experiment mit MAX17043 zur Ermittlung des Ladezustandes . . .	55
3.4	Abstrakte Regelungsschleife des Entlademanagements . . . . .	60
3.5	Systemverhalten mit Unterschreitung der Schranke . . . . .	61
3.6	Verbleibende Energie bei verschiedenen Parameterkombinationen . .	65
3.7	Management-Experiment mit einem 1F Superkondensator . . . . .	67
3.8	Management-Experiment mit 550mAh Akkumulator . . . . .	69

3.9	Management-Experiment mit 80% geladenem Akkumulator . . . . .	72
3.10	Management-Experiment mit CR2032 Knopfzelle . . . . .	74
3.11	Management-Experiment mit AAA Batterien . . . . .	75
4.1	Anwendungssteuerung unter Einbeziehung von Verbrauch und Batterie	79
4.2	Last und Verbrauch durch Geräteaktivitäten . . . . .	80
4.3	Mögliche Granularitäten der Erfassung des $\mu\text{C}$ . . . . .	88
4.4	Beispiel: Einfluss der Last auf die Effizienz eines Spannungsreglers .	89
4.5	Mehraufwand der Verbrauchserfassung . . . . .	93
4.6	Mehraufwand bei Anpassung an Spannungskonverter . . . . .	94
4.7	Abweichung der Modelle von der gemessenen Stromaufnahme . . . .	95
4.8	Modelle mit hoher Auflösung und per-Knoten-Kalibrierung . . . . .	96
4.9	Beitrag der Geräte/Modi zum gesamten Verbrauch für jedes Modell	97
4.10	Dynamisch berechnete Anwendungsfrequenz . . . . .	100
4.11	Verwendender Netzwerkaufbau . . . . .	100
4.12	Wege der Nachrichten innerhalb von 60 Stunden nach Absender . . .	101
4.13	Stromaufnahme nach 60 Stunden . . . . .	102
4.14	Stromaufnahme des Sendemodus über 60 Stunden . . . . .	102
5.1	Anwendungssteuerung mittels generischen Managements . . . . .	105
5.2	Schematische Darstellung des Verteilungsproblems . . . . .	107
5.3	Anpassung des Messintervalls im Sensornetz . . . . .	122
5.4	Ereignisfluss einer vereinfachten Sensornetzanwendung . . . . .	125
5.5	Beispiel für die faire Abrechnung einer geteilten Ressource . . . . .	127
5.6	Typische Nutzenfunktionen für den Energieeinsatz . . . . .	129
5.7	Strategien bei Energiemangel . . . . .	132
5.8	Strategien für Minderbedarf . . . . .	133
5.9	Dynamische Berechnung des Messintervalls . . . . .	137
5.10	Mehraufwand des Energiemanagements mit einer Funkanwendung .	142
5.11	Laufzeit der Energieverteilung auf MSP430 . . . . .	143
5.12	Energieverteilungsheuristiken . . . . .	144
5.13	Auswirkung von Gewichten bei Budgetzahl . . . . .	145
5.14	Verteilung bei unzureichender Energieverfügbarkeit . . . . .	146
5.15	Einfluss des limitierten Leihens . . . . .	147
5.16	Einfluss des Gebers auf das Leihen . . . . .	148
5.17	Einfluss des Stehlens . . . . .	149
5.18	Kombination von Leihen und Stehlen . . . . .	150
5.19	Limitierung des Ansparens . . . . .	151
5.20	Auswirkungen der Verteilung des tatsächlichen Bedarfs . . . . .	152
5.21	Auswirkungen der Adaption . . . . .	153

5.22	Experimenteller Vergleich von Managementansätzen . . . . .	156
6.1	Szenario für die Überwachung von Geräten . . . . .	160
6.2	Hinweissignale . . . . .	162
6.3	Durchschnittliches Anwendungsintervall . . . . .	162
6.4	Verteilung der Ereignisse . . . . .	163
6.5	Spannungsverlauf mit und ohne Management . . . . .	163
6.6	Auswirkungen der Ereignisse . . . . .	164
6.7	Überblick über das Wassereinzugsgebiet Hühnerwasser . . . . .	165
6.8	Zonen mit unterschiedlicher Vegetationsdichte . . . . .	166
6.9	Durchschnittliche Regenmenge . . . . .	168
6.10	Gemessene Temperatur und Luftfeuchte . . . . .	169
6.11	Dynamischer Messintervall . . . . .	169
6.12	Zuteilung zu den Budgets . . . . .	170
6.13	Stromaufnahme der Knoten . . . . .	171
6.14	Batteriespannung der Knoten . . . . .	171
7.1	Entwickelter Energiemanagementansatz . . . . .	173
A.1	Verbleibende Energie bei verschiedenen Parameterkombinationen . .	178
A.2	Limitierung der kleinstmöglichen oberen Schranke . . . . .	179





# Tabellenverzeichnis

2.1	Eigenschaften gebräuchlicher Primär- und Sekundärzellen . . . . .	11
2.2	Energiegewinnung aus der Umwelt . . . . .	18
2.3	Vergleich verschiedener Mikrocontroller . . . . .	22
2.4	Vergleich verschiedener Funkmodule . . . . .	28
2.5	Vergleich von Medienzugriffsprotokollen . . . . .	32
2.6	Vergleich der Stromaufnahme verfügbarer Sensoren . . . . .	35
2.7	Gemessene Betriebsmodi des MSP CC430F6137 . . . . .	39
2.8	Stromaufnahme vom Beispielknoten . . . . .	47
3.1	Bewertung der verschiedenen Ansätze . . . . .	59
3.2	Parameter des 5. Experiments . . . . .	73
4.1	Bewertung der verschiedenen Ansätze . . . . .	87
4.2	Verbrauchsmodelle auf Grundlage der ermittelten Stromaufnahme .	91
4.3	Speicherbedarf für einzelne Komponenten der Verbrauchserfassung .	92
4.4	Vergleich der Messung mit der Verbrauchserfassung . . . . .	98
4.5	Vergleich der Messung mit der Verbrauchserfassung für Spannungsregler	99
5.1	Bewertung der verschiedenen Ansätze . . . . .	119
5.2	Speicherbedarf der Komponenten des Energiemanagements . . . . .	141
5.3	Verwendete Budgetanforderungen . . . . .	144
5.4	Bewertung der Strategien zur Anpassung an Bedarfsänderungen . .	154
6.1	Parameter des Kühlschrankszenarios . . . . .	161
6.2	Parameter des Umweltmonitoringszenarios . . . . .	167
B.1	Auswertung der gemessenen Stromaufnahme . . . . .	182
B.2	Pakete pro Sekunde im Ping-Pong-Verfahren mit Verbrauchserfassung	183
B.3	Pakete pro Sekunde mit Spannungsregler . . . . .	183
B.4	Pakete pro Sekunde im Ping-Pong-Verfahren mit Energiemanagement	183



# Symbolverzeichnis

## Kapitel 3

$T$	Anzahl der Zeitschritte
$t$	Aktueller Zeitschritt
$V_{cutoff}$	Abschaltspannung
$V_m(t)$	Gemessene Spannung in Zeitschritt $t$
$V_{ref}(t)$	Berechnete Spannung in Zeitschritt $t$
$U(t)$	Obere Schranke
$L(t)$	Untere Schranke
$B+$	Konstante zur Berechnung der oberen Schranke
$B-$	Konstante zur Berechnung der unteren Schranke
$\Delta V(t)$	Berechneter Spannungsabfall im Zeitschritt $t$

## Kapitel 5

$\Phi$	Energieanteil
$E_{avail}$	Zur Verfügung stehende Energie
$t_{remain}$	Verbleibende Laufzeit
$b$	Energiebudget
$B$	Budgetanforderung
$[t_{start}, t_{end}]$	Gültigkeitsintervall
$min$	Minimaler Bedarf assoziierter Verbraucher
$max$	Maximaler Bedarf assoziierter Verbraucher
$g$	Gewicht
$U$	Nutzenfunktion
$E$	Im Intervall verfügbare Energie
$G$	Summe der Gewichte nicht gefüllter Budgets
$r$	Von assoziierten Verbrauchern angefragte Energie
$z$	Kreditrate
$s$	Geliehene Energie
$\widehat{min}$	Zeitabhängiger minimaler Bedarf
$\widehat{max}$	Zeitabhängiger maximaler Bedarf

$d$	Abgelaufene Zeit innerhalb eines Zeitschrittes
$D$	Dauer eines Zeitschrittes
$C$	Dynamische Begrenzung der speicherbaren Energie
$\kappa$	Prozentualer Anteil zur Berechnung der Begrenzung
$\tau$	Dynamisch berechnetes Anwendungsintervall
$c$	Energieverbrauch für die Ausführung der Anwendung
$\sigma$	Dynamisch berechnete Energiestufe
$N$	Anzahl der Servicestufen
$\gamma$	Berechnete Füllrate des Token Buckets
$tb$	Füllstand des Token Buckets
$t_{full}$	Seit dem letzten Füllen des Token Buckets verstrichene Zeit
$E_{soll}$	Ziel-Füllstand des Token Buckets zur Benachrichtigung der Anwendung

# Kapitel 1

## Einleitung

In unserem Alltag sind wir umgeben von eingebetteten Systemen. Für jedermann erkennbar oder im Hintergrund verrichten sie in Form von einfachsten Mikrocontrollern bis hin zu Mehr- und Multikernprozessoren ihre Arbeit. Vielen dieser Systeme ist gemein, dass sie nicht an das Stromnetz angeschlossen sind. Für diese mit Batterien oder Akkumulatoren betriebenen Geräte ist es notwendig, mit der beschränkten Ressource Energie zu haushalten. Dies gilt auch für Systeme, die Energie aus der Umgebung gewinnen können, da sie Zeiten ohne ausreichende Gewinnung überbrücken müssen. Ein verfrühter Ausfall wegen Energiemangel kann zum Beispiel bei einem Entertainmentsystem ärgerlich sein oder sogar lebensbedrohliche Folgen haben, wie beim Ausfall eines drahtlosen Feuermelders.

Ein Bereich, in dem das Energieproblem eine besondere Rolle spielt, sind die drahtlosen tief eingebetteten Systeme. In diese Kategorie fallen insbesondere die aus der Vision des intelligenten Staubs (Smart Dust) [1] hervorgegangenen drahtlosen Sensornetzwerke (wireless sensor networks - WSN) [2], deren Abkömmlinge, die körpernahen drahtlosen Netzwerke (wireless body-area networks - WBAN) [3], und das verwandte Internet der Dinge (Internet of Things - IoT) [4].

Während ein WBAN - und je nach Sichtweise auch ein lokales IoT - nur aus wenigen heterogenen Knoten besteht, setzt sich ein Sensornetzwerk aus einzelnen, meist homogenen Knoten zusammen, deren Anzahl in die Tausende gehen kann. Diese kleinen, aus günstiger Hardware bestehenden, tief eingebetteten Systeme kooperieren in großer Zahl miteinander und werden in verschiedensten Anwendungsszenarien eingesetzt. Dazu zählen die Überwachung der Umwelt z. B. [5, 6], die Präzisionslandwirtschaft z. B. [7, 8], die Tierbeobachtung z. B. [9, 10] und die Strukturüberwachung von Gebäuden und Brücken z. B. [11]. Sie dienen der autonomen Steuerung wie in [12] und werden für logistische z. B. [13, 14] und militärische Zwecke z. B. [15, 16] eingesetzt.

Die schiere Anzahl an Knoten macht es unmöglich, die Batterien in einem solchen Netz in kurzen Abständen zu tauschen. Dieses Problem ist für Netze, die in abgelegenen Regionen ausgebracht wurden, noch größer, da dort möglicherweise Expeditionen langfristig geplant werden müssen. Der Ausfall eines einzelnen Knotens kann dabei, abhängig von der Anwendung und der Definition der Netzwerklebenszeit [17], die Funktion des gesamten Netzwerkes einschränken oder sogar verhindern.

Auch für WBAN, die aus nur wenigen Geräten bestehen, ist ein verfrühter Ausfall problematisch. Diese Geräte besitzen bauartbedingt nur eine kleine Energiequelle, sollen im medizinischen Bereich aber mehrere Jahre lang arbeiten, wobei keinerlei Redundanz vorhanden ist [18]. Selbst in weniger kritischen Bereichen kann ein Ausfall problematisch sein und sollte vermieden werden. So wird ein Sportler über den Ausfall seines Fitnessmessgerätes, das Daten an sein Smartphone sendet, nicht erfreut sein und könnte wohl eher mit einer verringerten Datenrate leben.

Diese Art der Systeme unterscheidet sich von anderen eingebetteten drahtlosen Systemen, da Entscheidungen weitgehend autonom getroffen werden müssen. So kann zum Beispiel ein Smartphone seinem Nutzer detaillierte Entscheidungen überlassen. In der Regel werden diese auch zeitnah getroffen. Im Gegensatz dazu ist dies bei Sensornetzwerken oftmals mit hoher Latenz verbunden und bei einer großen Anzahl an Knoten nicht mehr praktikabel. Sie müssen weitgehend autonom agieren.

Es existieren verschiedene hard- und softwarebasierte Ansätze, um den Energieverbrauch eingebetteter Systeme zu senken. Dadurch wird es ermöglicht, die Laufzeit zu verlängern. Auf der Ebene der Hardware ist Energiesparen allein allerdings kein Allheilmittel, denn es geht immer mit einer Verringerung der von der Hardware geleisteten Arbeit einher. Ein extremer Fall wäre es, den Energieverbrauch durch Verzicht auf Arbeit stark zu verringern und so eine hohe Lebenszeit zu erreichen. Dies ist aber weder sinnvoll noch erstrebenswert. Es muss ein Kompromiss zwischen den minimalen Anforderungen der Anwendung und dem maximal technisch Möglichen gefunden werden.

Mithilfe von sanfter Verschlechterung (engl. graceful degradation) lässt sich die Laufzeit eines Systems adaptiv verlängern. Dabei wird an bestimmten Punkten die Leistung des Systems verringert, zum Beispiel anhand verbleibender Batteriekapazität, wodurch gleichzeitig an Laufzeit gewonnen wird. Nach und nach verliert das System so an Fähigkeiten. Der Vorteil eines solchen Vorgehens ist, dass die maximale Leistung solange wie möglich aufrecht erhalten wird und nur allmählich abnimmt. Dadurch eignet es sich für Szenarien, in denen vom Nutzer zwar eine hohe Leistung bevorzugt wird, die Anforderungen jedoch nicht fix sind und einen Leistungsbereich umfassen. Bei drahtlosen eingebetteten Systemen wird dieser Leistungsbereich oft durch die zulässige Rate der Kommunikation, der Datenerfassung und die Messauflösung umfasst.

---

Das Erreichen einer möglichst hohen Laufzeit ist jedoch nicht in jedem Fall entscheidend. Für einen Teil der Anwendungen ist es notwendig, eine bestimmte Laufzeit zu erreichen, weil z. B. fixe Wartungsintervalle eingehalten werden müssen oder die Knoten in unwegsamem Gelände ausgebracht wurden und schwer erreichbar sind. Die naheliegende Lösung, den Energiespeicher durch Überabschätzung des Verbrauchs zu dimensionieren oder Energiegewinnung aus der Umgebung zu nutzen, ist nicht in jedem Fall praktikabel. Eine worst-case-Abschätzung kann durch die Wirtschaftlichkeit oder durch ein begrenztes Raumangebot und/oder Gewichtsvorgaben eingeschränkt sein. Die Leistung des Systems muss also von vornherein verringert sein oder durch graceful degradation angepasst werden, um eine ausreichende Überabschätzung zu ermöglichen, sodass ein Unterschreiten der geplanten Laufzeit verhindert wird. Das Erreichen eines definierten Lebenszeitziels hat somit Vorrang vor Einbußen hinsichtlich der Anwendungsqualität [19]. Kann aus einer längeren Laufzeit kein Vorteil gewonnen werden, ist alle bei Erreichen des Ziels überschüssige Energie verschwendet. Ungenutzte Energie ist ein Indikator dafür, dass potenziell mehr Arbeit und somit eine höhere Anwendungsqualität (sei es nun eine höhere Datenrate oder umfangreichere Berechnungen) möglich gewesen wäre.

Um einem verfrühten Ausfall zu begegnen, ist es notwendig, die aktuell zur Verfügung stehende Energiemenge und den Verbrauch zu kennen, um daraus die verbleibende Laufzeit zu berechnen. Zusammen mit den Lebenszeitvorgaben kann ermittelt werden, ob das System Energie sparen muss oder den Verbrauch erhöhen kann. Die Möglichkeit der Adaption ist hierfür unerlässlich. Wenn der Verbrauch nicht durch das System angepasst werden kann, ist das Erreichen von Zielen nur durch externe Faktoren bestimmt und kann nicht vom System selbst beeinflusst werden.

Es ergeben sich drei Fragen, die durch ein Energiemanagement beantwortet werden müssen, das Lebenszeitziele zu erreichen sucht:

1. Wie viel Energie steht zur Verfügung?
2. Wie hoch ist der Energieverbrauch?
3. Wie wird die zur Verfügung stehende Energie unter den Verbrauchern<sup>1</sup> aufgeteilt, sodass die Anforderungen der Anwendung erfüllt werden?

Abhängig von der genutzten Energiequelle ist die Abschätzung der verbleibenden Energie kompliziert. Batterien basieren auf chemischen Vorgängen und unterliegen nichtlinearen Effekten, die sich auf die verbleibende Kapazität auswirken. Zusätzlich sind sie gegenüber äußeren Einflüssen anfällig, insbesondere gegenüber der Temperatur [20]. Dies macht komplizierte chemische/statistische Modelle beziehungsweise spezialisierte Hardware notwendig, um akkurate Aussagen über den

---

<sup>1</sup>Verbraucher bezieht sich hierbei auf Hard- und Softwarekomponenten eines einzelnen eingebetteten Systems

Ladezustand treffen zu können. Existierende Ansätze leiden unter ihrer Komplexität, Kosten und/oder Abhängigkeiten von speziellen Batterieherstellern und -typen. Ein Indiz für die verbleibende Restenergie von Batterien ist die Spannung. Jeder Batterietyp weist in Abhängigkeit von Last und Temperatur ein charakteristisches Entladeprofil auf. Dieses kann genutzt werden, um mithilfe aufgezeichneter Kurven auf die verbleibende Energie zu schließen. Allerdings können sich Chargin desselben Batterietyps und -herstellers unterschiedlich verhalten, was in [21] dazu führte, dass das System nicht in der Lage war, dass vorgegebene Lebenszeitziel zu erreichen. Bei auf Energiegewinnung basierenden Systemen kommt hinzu, dass sie eine Vorhersage über die zu erwartende Menge an gewonnener Energie treffen müssen, um eine kontinuierliche Lauffähigkeit zu ermöglichen [22].

Eine Prognose des Energieverbrauchs vor dem Einsatz ist oftmals schwierig, da ein genaues Anwendungsverhalten nicht immer vorhersehbar ist. Dies kann durch die explizite Nutzung adaptiver Datenerfassung und dem damit einhergehenden variablen Datenaufkommen gewollt sein oder durch von Störungen ausgelöste Retransmissionen bei der Kommunikation ungewollt auftreten. Aber auch wenn das Anwendungsverhalten exakt vorhersagbar ist, können Fertigungsvarianzen und -fehler der Hardware einen veränderten Verbrauch verursachen, ohne den Betrieb zu verhindern. Selbst auf demselben Knoten kann der Verbrauch variieren, zum Beispiel aufgrund sinkender Batteriespannung oder der von der Belastung abhängigen Effizienz eines Spannungskonverters. Eine Erfassung des Verbrauchs zur Laufzeit ist daher Voraussetzung, um die Aktivitäten zu steuern und Lebenszeitziele zu erreichen.

Eine Aufteilung der verfügbaren Energie unter den Verbrauchern macht es notwendig, den Verbrauch für jeden dieser Verbraucher einzeln zu erfassen. Dies ermöglicht es, den Verbrauchern ein Quantum Energie zuzuweisen und dabei Aktivitäten nur dann zu erlauben, wenn ausreichend Energie vorhanden ist. Daraus folgt eine logische Isolation der Verbraucher untereinander. Jeder Verbraucher kann so potenziell selbst mit seiner Energie haushalten. Eine solch strikte Handhabung ist jedoch nicht immer sinnvoll. Verbraucher sollten kooperieren, um die Anwendungsqualität auch in Sonderfällen aufrecht zu erhalten und nicht genutzte Energie sinnvoll zu verwenden.

Varianzen, die sowohl auf der Ebene der Energieversorgung als auch bei der Hard- und Software eines drahtlosen tief eingebetteten Systems existieren, muss ein Energiemanagement auf allen Ebenen und ebenenübergreifend berücksichtigen. Da eine Vorhersage der zur Verfügung stehenden Energie und des Verbrauches durch die Varianzen nur grob möglich ist, ist ein dynamischer Ansatz beim Erreichen von Lebenszeitzielen leistungsfähiger als eine rein statische Lösung. Statische Ansätze müssen letztlich pessimistisch vorgehen, was sich in gesteigerten Kosten durch größere Batterien äußert. Durch die Überwachung der Energie- und Verbrauchsparameter



während des Betriebs ist ein dynamisches Energiemanagement in der Lage, die zur Verfügung stehende Energie besser auszunutzen. Dabei muss der dazu notwendige Mehraufwand hinsichtlich Rechenleistung und damit Energie gering bleiben, um das Kosten-Nutzen-Verhältnis zu wahren.

### 1.1 Beitrag der Arbeit

In dieser Arbeit wird ein Energiemanagement vorgestellt, das dynamische Einflüsse auf den Energieverbrauch und die Versorgung zur Laufzeit einbezieht. Ist die Anwendung in der Lage, durch Anpassung mindestens eines Leistungsparameters (Messzyklus, Auflösung, Sendestärke und -häufigkeit etc.) auf eine veränderte Energieverfügbarkeit zu reagieren, kann die bei statischem Management notwendige Überabschätzung des Verbrauchs verringert werden. Sonst verschwendete Energie wird so unter Beibehaltung des Lebenszeitziels nutzbar gemacht. Falls notwendig wird die Leistung des Systems verringert, um den Energieverbrauch zu senken und die Laufzeit zu erhöhen. Zur Beantwortung der drei in Kapitel 1 vorgestellten Fragen werden existierende Verfahren und Ansätze erweitert, um diese gegen die Varianzen abzu härten.

Der Zustand der Batterien wird mithilfe einfacher Linearisierung gemessener Spannungswerte abgeschätzt. Ausgehend von Arbeiten in [23] werden grobe Batteriecharakteristika einbezogen. Dem Energiemanagement wird durch ein Hinweissignal mitgeteilt, ob Energie gespart werden muss oder der Verbrauch erhöht werden kann. Im Gegensatz zu anderen Ansätzen wird keine zusätzliche Hardware benötigt und auf komplexe Parameter oder Berechnungen verzichtet.

Die Ermittlung der verbrauchten Energiemenge erfolgt durch ein rein Softwarebasiertes Verfahren. Dazu werden Zustandsübergänge der Hardware observiert. Dies ermöglicht eine feingranulare Erfassung und flexible Zuordnung des Verbrauchs. Durch Fertigungstoleranzen verursachten Unterschieden wird mittels statistische Modellierung des Verbrauchs begegnet. Varianzen in der Stromaufnahme durch unterschiedliche Versorgungsspannungen und etwaige Konverter können im Gegensatz zu gleichartigen Ansätzen zur Laufzeit einbezogen werden.

Das Management nutzt sogenannte Energiebudgets, die das Prinzip der „Resource Container“ [24] umsetzen. Anwendungs- und Systemteilen wird ein Recht auf Energie zugewiesen, deren Menge von den individuellen Verbrauchsgrenzen, etwaigen Prioritäten und der verfügbaren Energie bestimmt wird. Die Analyse existierender Ansätze in Kapitel 5.2 zeigt, dass auf kurz- und langfristige Bedarfsänderungen zur Laufzeit bisher nur unzureichend eingegangen wird. In dieser Arbeit werden deshalb die Auswirkungen solcher Änderungen und Strategien zu ihrer Behandlung betrachtet. Energiebudgets können flexibel zu Anwendungsteilen, Geräten

und Ausführungspfaden zugeordnet werden. Das vorgestellte Management ermöglicht eine Kombination aus Strategien existierender Ansätze, um eine Anpassung des Verbrauchs seitens der Anwendung an eine veränderte Energieverfügbarkeit zu erreichen. Anwendungen können sowohl bewusst auf das ihnen zugewiesene Budget reagieren, als auch bereitgestellte Mechanismen zur energieabhängigen Berechnung von Anwendungsintervallen und Servicestufen nutzen.

Der Nachweis der Wirksamkeit dieses dynamischen Energiemanagementansatzes erfolgt durch eine prototypische Umsetzung für zwei Anwendungsfälle. Beide Szenarien unterscheiden sich hinsichtlich Laufzeit und Datenaufkommen, setzen aber das Einhalten eines strikten Wartungsintervalls voraus.

Beim ersten Szenario wird die Funktion einer Maschine durch mehrere Systeme überwacht, um Rückschlüsse auf die Funktionsfähigkeit ziehen zu können. Diese senden ihre Daten direkt an eine Zwischenstation zur Weiterleitung. Die Erfassungsrate wird anhand der verfügbaren Energie bestimmt. Bewegungssensoren stellen fest, ob Ereignisse vorliegen. In diesem Fall wird die Datenerfassungs- und Senderate für einen Zeitraum erhöht.

Das zweite Szenario umfasst eine typische Umweltüberwachung aus dem Sensor-netzbereich. An verschiedenen Punkten werden periodisch Daten und spontane Ereignisse erfasst und durch das Netz an eine Basisstation gesendet. Um Datenverlust zu verhindern, werden alle Daten auf dem Knoten gespeichert und nach Ende des Experimentes ausgelesen. Die erfassten Daten sind von unterschiedlicher Relevanz, wodurch sich, abhängig von der verfügbaren Energie, unterschiedliche Erfassungsraten ergeben.

## 1.2 Zugrundeliegende Veröffentlichungen

Diese Arbeit baut auf folgenden Veröffentlichungen auf:

- SIEBER, André ; NOLTE, Jörg: Utilizing Voltage Decline for Reaching Lifetime Goals. In: *10. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze, Forschungsbericht TR-RI-11-313*. Universität Paderborn, 2011, S. 57-60
- SIEBER, André ; NOLTE, Jörg: Datasheet vs. Real World: A Look on Sensor Node Energy Consumption. In: *Proceedings of the IEEE International Conference on Green Computing and Communications*. Besançon, Frankreich, 2012, S. 641 - 643
- SIEBER, André ; NOLTE, Jörg: Online device-level energy accounting for wireless sensor nodes. In: *Proceedings of the 10th European conference on Wireless Sensor Networks*. Ghent, Belgien, 2013, S. 149-164

- SIEBER, André ; KARNAPKE, Reinhardt ; NOLTE, Jörg: From Energy Accounting to Energy Management. In: *13. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze, Forschungsbericht TR-2014-1*. Universität Potsdam, 2014, S. 25-28
- SIEBER, André ; NOLTE, Jörg ; KARNAPKE, Reinhardt: Using Energy Budgets to Reach Lifetime Goals while Compensating Dynamic Effects. In: *Proceedings of the ninth International Conference on Sensor Technologies and Applications*. Neapel, Italien, 2015, S 7-13
- SIEBER, André ; NOLTE, Jörg ; KARNAPKE, Reinhardt: Compensating Dynamic Effects with Budget-based Energy Management to Reach Lifetime Goals. In: *Sensors & Transducers* 194 (2015), Nr. 11, S. 22-34

## 1.3 Struktur der Arbeit

Der Rest dieser Arbeit gliedert sich wie folgt: In Kapitel 2 wird auf die Erzeugung und den Verbrauch von Energie in drahtlosen eingebetteten Systemen eingegangen. Ansätze zur Energieeinsparung und ihre Auswirkungen werden diskutiert. Für eine Plattform werden die Hardwarevarianzen durch Messung gezeigt. Kapitel 3 befasst sich mit verschiedenen Ansätzen zur Erfassung des Ladezustandes von Batterien. Es wird ein Hinweis-basiertes Entlademanagement vorgestellt, das das Erreichen von Lebenszeitzielen durch Erfassung der Batteriespannung ermöglicht. Kapitel 4 betrachtet die feingranulare Erfassung des Energieverbrauchs während der Laufzeit. Ausgehend von Messergebnissen aus Kapitel 2 wird ein Software-basiertes Verfahren entwickelt und evaluiert. Der Ansatz bezieht Varianzen ein, die durch die Nutzung von Spannungswandlern und von spannungssensiblen Geräten entstehen. Basierend auf den Informationen des Batteriemanagements und der Erfassung des Energieverbrauchs wird in Kapitel 5 das Energiemanagement vorgestellt. Durch Zuteilung von Energiebudgets zu Hard- und Softwareaktivitäten können mithilfe von Isolation und Kooperation Lebenszeitziele erreichen werden. In Kapitel 6 erfolgt die Demonstration des Gesamtsystems mithilfe von zwei Anwendungsfällen. Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick in Kapitel 7.



## Kapitel 2

# Energieversorgung und -verbrauch in drahtlosen tief eingebetteten Systemen

Dieses Kapitel gibt einen Überblick über Energieversorgung und Energieverbrauch in drahtlosen tief eingebetteten Systemen. Zu diesem Zweck werden grundlegende Technologien und Ansätze vorgestellt.

Der Energieverbrauch eines Systems bestimmt sich aus der elektrischen Leistung, integriert über die Zeit. Die Leistungsaufnahme errechnet sich aus der Multiplikation der Spannung mit der Stromstärke ( $P = U * I$ ) und hängt von den aktiven Hardwarekomponenten ab. Welche Hardwarekomponenten aktiv sind, steht gewöhnlich unter der Kontrolle der Software. Der Verbrauch einer Komponente lässt sich meist aus dem Datenblatt des Herstellers ablesen. Dort werden Informationen zum Verbrauch einer Komponente unter verschiedenen Bedingungen aufgelistet. Aber nicht jede Komponente und jedes System sind gleich, Varianzen und Toleranzen können signifikante Unterschiede hervorrufen.

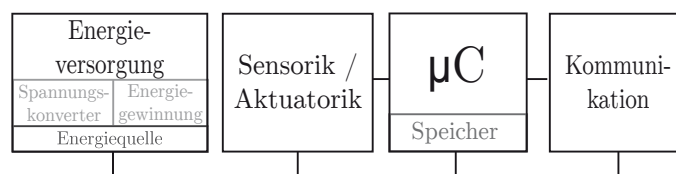


Abbildung 2.1: Typischer Aufbau eines drahtlosen tief eingebetteten Systems

Der typische Aufbau eines drahtlosen tief eingebetteten Systems ist in Abbildung 2.1 dargestellt. Es besteht aus vier wesentlichen Komponenten. Die Energieversorgung besteht mindestens aus einer Energiequelle, kann aber auch Spannungskonverter oder Hardware für die Energiegewinnung (engl. Harvesting) enthalten. Letztere steht jedoch nicht im Fokus dieser Arbeit.

Sensoren dienen der Erfassung physikalischer Parameter, die gesammelt, verarbeitet und weitergeleitet werden. Je nach Anwendungsszenario ist neben der Erfassung von Umweltparametern auch die Manipulation der Umwelt durch Aktuatoren möglich. Für die Kommunikation werden Funkmodule eingesetzt. Der Mikrocontroller ( $\mu\text{C}$ ) übernimmt mit seinem Prozessor die Steuerung des Systems und besitzt integrierten RAM/ROM und/oder greift auf einen vorhandenen externen Speicher zu.

Für jede dieser Komponenten existiert eine Vielzahl an Möglichkeiten, die unterschiedlich leistungsfähig sind, unterschiedliche Kosten und einen unterschiedlichen Stromverbrauch verursachen. Jede Komponente kann in unterschiedlicher Ausprägung und Zahl in einem System vorhanden sein. Neben diesen Komponenten besteht ein drahtloses tief eingebettetes System noch aus weiteren digitalen und analogen Bauteilen, die den Stromverbrauch beeinflussen können.

In diesem Kapitel wird dargestellt, nach welchem Muster die Komponenten eines solchen Systems mit Energie versorgt werden, sie verbrauchen und in welchem Zusammenhang der Verbrauch mit der Energieversorgung steht. Dabei wird auch auf mögliche fertigungs- und umweltbedingte Varianzen und deren potenzielle Auswirkungen eingegangen.

## 2.1 Energieversorgung

Abhängig von der Anwendung sollen drahtlose tief eingebettete Systeme Standzeiten von mehreren Jahren erreichen. Voraussetzung dafür ist eine zuverlässige Energiequelle. Zur Speicherung von Energie existieren verschiedenste Technologien mit unterschiedlicher Leistungsfähigkeit. Primärzellen - nicht wieder aufladbare Batterien - sind kostengünstig und leicht verfügbar. Die wiederaufladbaren Sekundärzellen eignen sich für Szenarien, in denen Energie gewonnen und gespeichert werden kann. Dies gilt auch für die sogenannten Superkondensatoren, die im Vergleich zu normalen Kondensatoren eine hohe Speicherkapazität und Zyklenfestigkeit besitzen. Andere Energiespeicher wie Brennstoffzellen und Kleinstwärmekraftmaschinen sind besonders durch ihre hohe Energiedichte interessant, aber bisher nur in der Forschung thematisiert.

### 2.1.1 Primär- und Sekundärzellen

Primär- und Sekundärzellen sind galvanische Zellen, die durch Oxidation und Reduktion chemische Energie in elektrische Energie umwandeln. Im Gegensatz zu Primärzellen können Sekundärzellen durch Umkehrung des Prozesses wieder aufgeladen werden [20, S. 1.3]. Batterien und Akkumulatoren bestehen aus einer oder mehreren dieser Zellen. Die Wahl der verwendeten Materialien bestimmt die Eigenschaften der Zellen, dementsprechend gibt es eine Vielzahl unterschiedlicher Typen.

Tabelle 2.1: Eigenschaften gebräuchlicher Primär(P)- und Sekundär(S)-Zellen bei 20°C(Daten entnommen aus [20, Tab. 8.3, 15.2])

Technologie	Energiedichte Knopf-/Rundzelle Wh/kg	typische Zellen [25] Ah(Bauform)	typischer Spannungsbereich (nominal) V	Temperatur- bereich °C	Zyklus- festigkeit	Selbstentladung
Li/MnO <sub>2</sub> (P)	230/230	1,5( $\frac{2}{3}$ A) 0,24 (CR2032)	3,3-2,0 (3,0)	-20 bis 55	-	1-2 % (Jahr)
Zn/Alk/MnO <sub>2</sub> (P)	80/145	2,9(AA) 1,25(AAA)	1,6-0,9 (1,5)	-20 bis 55	-	4 % (Jahr)
		8(C) 20(D)				
Ni-MH (S)	75	2,2(AA)	1,4-1,0 (1,2)	-20 bis 50	300-600	15-25 % (Monat) <sup>1</sup>
Li-Ion (S)	150 <sup>2</sup>	2(A)	4,2-2,5 (3,7)	-20 bis 50	1000+	2-10 % (Monat)

<sup>1</sup> Ni-Mh mit geringer Selbstentladung (beispielsweise Panasonic eneloop [26]) erreichen einen Ladestand von bis zu 90% nach einem und 70% nach fünf Jahren

<sup>2</sup> Prototypen bis zu 400Wh/kg (Envia Systems, [www.enviasystems.com](http://www.enviasystems.com))

In Tabelle 2.1 sind die Charakteristiken der gebräuchlichsten Primär- (P) und Sekundärzellen (S) aufgelistet. Vergleicht man beide Typen, wird ein Unterschied in der Energiedichte und in der Selbstentladung deutlich. Insbesondere die Selbstentladung ist bei Sekundärzellen um ein Vielfaches höher. Abbildung 2.2 zeigt diesen Unterschied approximiert in Abhängigkeit von der Temperatur in einer invertierten logarithmischen Skala. Diese hohe Selbstentladung schränkt die Verwendungsmöglichkeiten von Akkumulatoren in drahtlosen tief eingebetteten Systemen ein. Hohe Standzeiten im Bereich mehrerer Jahre sind mit ihnen kaum realisierbar. Als Alternative bietet sich die Verwendung von Energiegewinnung aus der Umwelt und/oder die Nutzung spezialisierter Ni-MH Zellen mit geringerer Selbstentladung [26] an.

Sekundärzellen selbst unterscheiden sich zusätzlich hinsichtlich ihrer Zyklenfestigkeit und der benötigten Lademethode. Des Weiteren spielen das Alter und der Zustand der Zelle für die zur Verfügung stehende Kapazität eine Rolle.

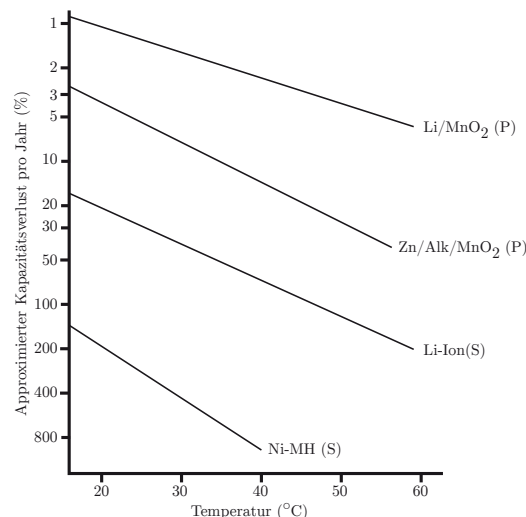


Abbildung 2.2: Einfluss der Temperatur auf die jährliche Selbstentladung gebräuchlicher Primär(P)- und Sekundär(S)-Zellen (nach [20, S. 32.5])

Die tatsächliche Kapazität von Batterien hängt nicht nur von der Technologie, sondern auch vom Hersteller und der Produktreihe ab. In [27] wurde experimentell die tatsächliche Kapazität mit jeweils zwei Entladeströmen für handelsübliche AA und AAA Batterien ermittelt. So unterscheidet sich die bei geringer Last ermittelte Kapazität bei Zn/Alk/MnO<sub>2</sub> (Alkaline) Batterien um bis zu 29% und bei hoher Last um bis zu 36%. Die Experimente zeigten zusätzlich, dass die tatsächliche Kapazität bei hoher Last deutlich niedriger ist als angegeben.

Variationen zwischen einzelnen Batterien desselben Herstellers und der gleichen Produktreihe wurden in [28] beobachtet und führten in [21] zu Problemen mit vorzeitigen Knotenausfällen, die erst durch Adaption an ein verändertes Entladeverhalten behoben werden konnten.



### Einfluss der Temperatur

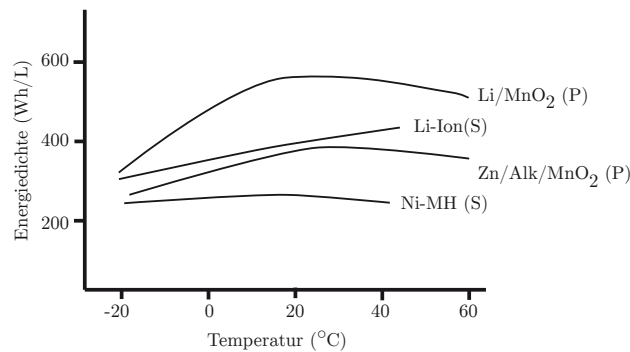


Abbildung 2.3: Einfluss der Temperatur auf die Energiedichte bei Primär-(P) und Sekundär-(S) Zellen (nach [20, S. 32.14])

Je nach verwendeten Materialien unterscheidet sich der Temperaturbereich, in dem eine Zelle arbeitet (siehe Tabelle 2.1). Dabei spielt die Temperatur für die zu erwartende Kapazität eine große Rolle [20, S 3.9]. Die Reaktionsgeschwindigkeit chemischer Prozesse nimmt bei steigender Temperatur zu. Eine Erhöhung der Temperatur um 10 Grad verdoppelt die Reaktionsgeschwindigkeit [29, S.151]. Abbildung 2.3 stellt die Energiedichte von gebräuchlichen Batterien und Akkumulatoren in Abhängigkeit von der Temperatur dar. Die Stärke des Einflusses unterscheidet sich je nach Technologie und muss bei der Auswahl in Abhängigkeit vom Anwendungsszenario in Betracht gezogen werden.

Wie stark sich schwankende Temperaturen auf den Spannungsverlauf auswirken, ist in Abbildung 2.4 zu sehen. Abgebildet ist das Ergebnis eines im Rahmen dieser Arbeit durchgeführten Experimentes. Ein Sensorknoten wurde von Ende Oktober bis

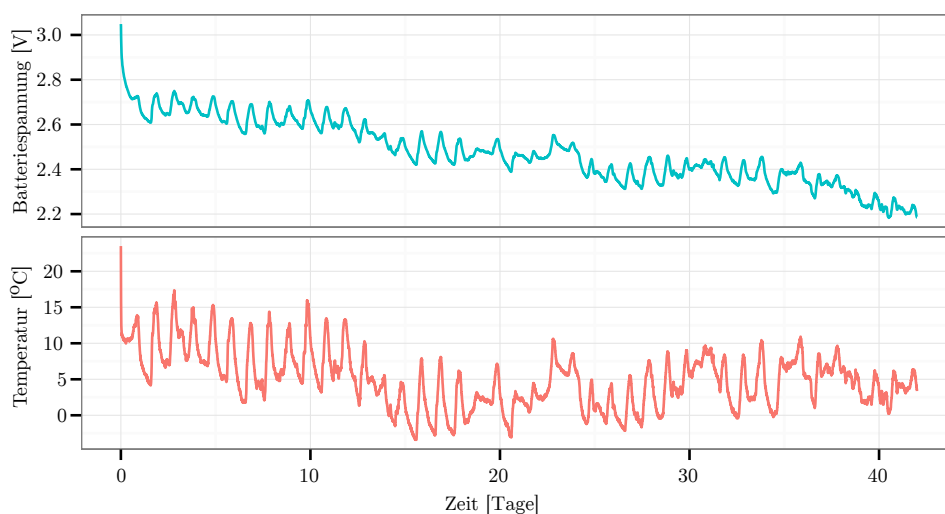
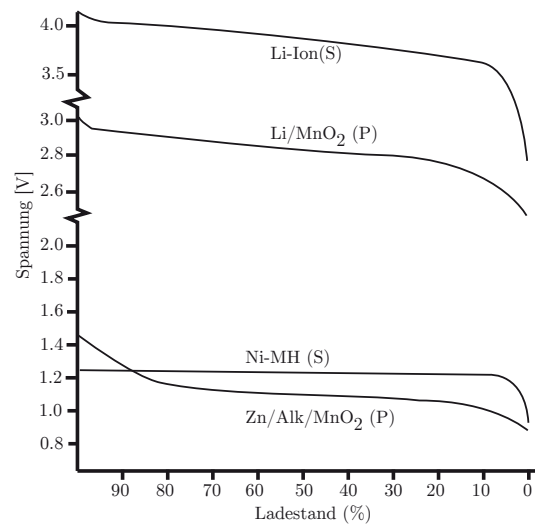


Abbildung 2.4: Einfluss der Außentemperatur auf die Spannung von zwei in Reihe geschalteten AA Alkaline Batterien von Ende Oktober bis Anfang Dezember 2011



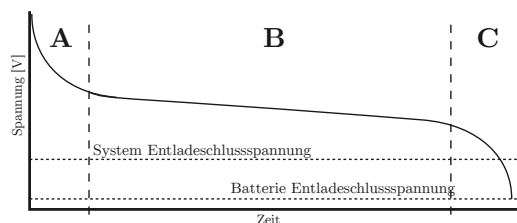
Abbildungung 2.5: Entladekurven von Primär-(P) und Sekundär-(S) Zellen (nach [20, S. 32.10])

Anfang Dezember 2011 im Freien ausgelegt (Fensterbank in Richtung Nordosten). Die Spannung der zwei in Reihe geschalteten No-Name-Batterien AA Zn/Alk/MnO<sub>2</sub> (Alkaline) und die Temperatur wurden periodisch ermittelt und per Funk zu einem PC übertragen. Jede Temperaturveränderung lässt sich in gedämpfter Form an der Spannung ablesen. Der Spannungsverlauf wirkt dadurch instabil.

### Spannungskurven

Abhängig von den verwendeten Materialien besitzen Primär- und Sekundärzellen eine charakteristische Spannungskurve [29, S. 111]. In Abbildung 2.5 sind die Kurven für gebräuchliche Technologien dargestellt. Zwar unterscheiden sich die Verläufe, insbesondere was den Spannungsbereich betrifft, aber die Ausprägungen des Spannungsabfalls ähneln sich.

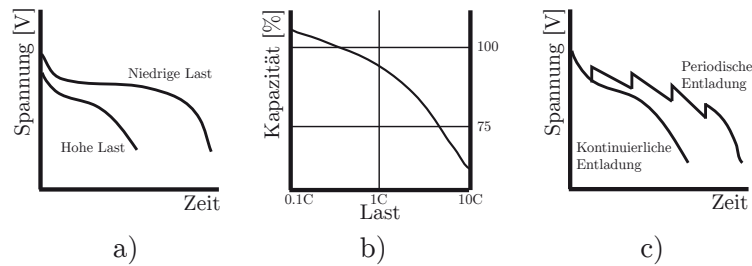
Wie in Abbildung 2.6 schematisch dargestellt, teilt sich für die meisten Zellen der Spannungsabfall in drei Teile. Deren Länge und Ausprägung hängt von der ver-



Abbildungung 2.6: Teile der Entladekurve: (A) schneller Spannungsabfall zur Nominalspannung, (B) nahezu lineare Kurve, (C) schneller Abfall zur Batterie Entladeschlussspannung

wendeten Technologie ab. In den Phasen A und C fällt die Spannung schnell ab. In Phase A nähert sie sich der Nominalspannung an und in Phase C fällt sie zur Entladeschlussspannung (engl. cut off voltage) der Zelle. Die Entladeschlussspannung bestimmt den Punkt, an dem technologiebedingt keine Energie mehr abgegeben werden kann. Abhängig von der angeschlossenen Hardware kann diese eine höhere Anforderung an die Spannung besitzen, sodass Energie nach Erreichen der System-Entladeschlussspannung ungenutzt bleibt. In [30] wird eine vierte Phase hinzugefügt, in der die Spannung sich durch den Erholungseffekt (engl. recovery effect) nach dem Abschalten der Last wieder erholt.

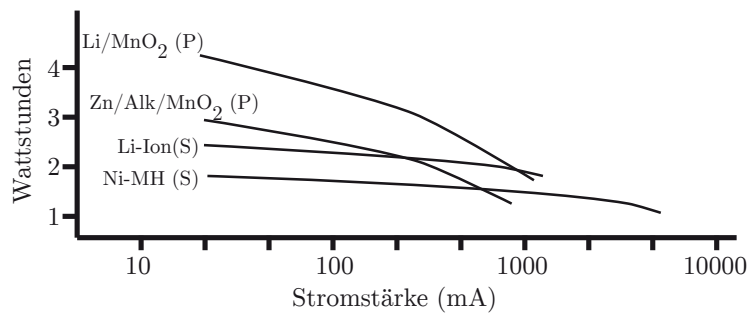
### Nicht-lineare Effekte



Abbildungung 2.7: Nicht-lineare Einflüsse auf die Kapazität a) Auswirkung der Last auf die Spannung (nach [20, S. 3.4]) b) Kapazitätsverlust bei steigender Last (nach [31]) c) Erholungseffekt durch periodische Entladung (nach [20, S 3.12])

Mehrere Faktoren bestimmen die tatsächlich aus einer Zelle entnehmbare Energie. Neben der Temperatur spielt das Nutzungsschema der Zelle eine Rolle. Wie in Abbildung 2.7 dargestellt, können hohe Entladeströme die Spannungskurve verändern (a) und die Kapazität verringern (b), während Erholungspausen eine Zelle regenerieren (c).

Der sogenannte *Peukert-Effekt* (engl. rate capacity effect) beschreibt die Abhängigkeit der entnehmbaren Energiemenge von der Höhe des Entladestroms [20, S 3.3].



Abbildungung 2.8: Auswirkung der Last auf AA (oder äquivalente) Primär-(P) und Sekundär-(S) Zellen (modifiziert entnommen aus [20, S. 32.13])

Abbildung 2.8 stellt die Leistung gebräuchlicher AA Batterien und Akkumulatoren der Stromstärke gegenüber. Es ist erkennbar, dass Primärzellen für diesen Effekt anfälliger sind.

Die in der Formel 2.1 beschriebene Peukert-Gleichung beschreibt diesen Zusammenhang zwischen Last und Kapazität [20, Kap. 3.2.6]. Dabei werden weder die chemischen noch die physikalischen Prozesse betrachtet, stattdessen stellt die Gleichung eine mathematische Näherung dar. Benötigt werden hierfür die Kapazität der Zelle bei einem Ampere Entladestrom  $C_p$ , die Entladezeit  $t$  und der Entladestrom  $I$ .

$$C_p = I^n * t \quad \text{bzw.} \quad t = \frac{C_p}{I^n} \quad (2.1)$$

Zusätzlich wird für jeden Typ von galvanischen Zellen ein sogenannter Peukert-Exponent  $n$  benötigt, der die Lastabhängigkeit beschreibt und experimentell bestimmt wird (zB. für Li-Ion 1,05 und 1,09 für NiMH Akkumulatoren). Andere Einflüsse werden durch die Gleichung nicht einbezogen, was die Genauigkeit begrenzt.

Meistens ist der Entladestrom nicht uniform, sondern wechselt in Dauer und Höhe. Dies kann zum Erholungseffekt (engl. recovery effect) führen, bei dem die Kapazität der Zelle nach einer Belastung wieder ansteigt und die Spannung sich regeneriert [20, S 3.12]. Die Stärke der Erholung ist sowohl von der Technologie als auch vom Lastgefälle abhängig.

Diese beiden nicht-linearen Effekte wurden bereits in der Forschung thematisiert. Der Einfluss von hohen, pulsformigen Entladeströmen auf Knopfzellen wurde auch in [32] und [33] untersucht. Beide zeigen, dass die Laufzeit negativ von einem erhöhten Entladestrom beeinflusst wird. In [33] wurde zusätzlich ein Einfluss der Batteriemarke nachgewiesen und zur Verminderung der hohen Entladeströme ein Kondensator als Puffer vorgeschlagen.

Der Einfluss der nicht-linearen Effekte von Batterien auf die Laufzeit von Sensorknoten wurde in [34] untersucht. Durch eine reduzierte Sendestärke auftretende Retransmissionen reduzieren die Laufzeit weniger stark, als das Senden mit hoher Stärke und weniger Retransmissionen. Zusätzlich wird eine dynamische Anpassung des Verbrauches bei niedrigen Temperaturen vorgeschlagen, um der geringeren Leistungsfähigkeit von Batterien bei niedriger Temperatur begegnen zu können.

In [35] wird ein Routingprotokoll vorgestellt, welches multiple Routen wählt, um den Peukert-Effekt zu reduzieren. Durch die Nutzung mehrerer Routen wird die Last verteilt und der Einfluss des Effekts reduziert. Dadurch werden höhere Laufzeiten möglich. Ein weiteres, in [36] vorgestelltes Protokoll wählt die Routen so, dass die Sendestärke minimiert werden kann, um so den Peukert-Effekt zu verringern. Zusätzlich wird die Last von Knoten mit geringem Batterieladestand reduziert, sodass diese vom Erholungseffekt profitieren können.

Der Erholungseffekt wurde in [37] experimentell untersucht. Dabei wurde festgestellt, dass der Hintergrundverbrauch, der durch den Schlafmodus des Systems verursacht wird, keinen Einfluss auf den Erholungseffekt hat. Es tritt ein Sättigungseffekt auf, welcher eine weitere Erholung trotz weiterer Verlängerung der Schlafperiode verhindert. Das Verhalten wurde mittels Markov-Ketten modelliert und ein Medienzugriffs- und Routing-Protokoll entworfen, das die Laufzeit unter Ausnutzung des Effekts verlängert.

### 2.1.2 Superkondensatoren

Die aus Doppelschichtkondensatoren weiterentwickelten Superkondensatoren zeichnen sich durch eine deutlich höhere Kapazität (bis zu mehreren 1000 F) als bei normalen Kondensatoren aus, wobei die hohe Zyklenfestigkeit erhalten geblieben ist. Im Vergleich zu Akkumulatoren ist diese um ein Vielfaches größer. Zwar ist ihre Energiedichte viel geringer, allerdings besitzen sie eine sehr viel höhere Leistungsdichte, können also sehr schnell ge- und entladen werden [38].

Die Entladekurve eines Superkondensators ist fast linear. Dadurch ergibt sich eine im Vergleich zu galvanischen Zellen bessere Relation zwischen Spannung und verbleibender Energie. Dabei ist der Spannungsverlauf nur wenig vom Entladestrom abhängig. Die Spannung bricht bei vergleichsweise höherer Last zu Beginn nur leicht ein.

Der Nachteil von Superkondensatoren ist ihre gegenüber Akkumulatoren noch einmal deutlich höhere Selbstentladung. Diese liegt im Bereich von 15 bis 45% pro Tag, wobei der Leckstrom mit zunehmender Kapazität ansteigt. Dies ergibt für große Superkondensatoren einen Verlust, der vergleichbar mit dem Verbrauch eines sehr sparsamen und selten aktiven drahtlosen tief eingebetteten Systems ist [39].

Diese Eigenschaften prädestinieren Superkondensatoren für Szenarien, in denen der Energiespeicher in kurzen Abständen geladen werden kann, die Selbstentladung also nebensächlich ist.

### 2.1.3 Energiegewinnung aus der Umwelt

Lange Standzeiten und ein vergleichsweise hoher Energieverbrauch schließen sich meistens gegenseitig aus. Aber auch ein geringer Verbrauch kann das Erreichen eines Lebenszeitziels nicht in jedem Fall garantieren, da Anforderungen der Anwendung an die Aktivität des Systems dem entgegen stehen können. Eine Erhöhung der Energiekapazität ist oft aufgrund von Beschränkungen in der Baugröße und in finanzieller Hinsicht limitiert.

Ein Ausweg aus dieser Situation ist die Energiegewinnung aus der Umwelt, die theoretisch einen aus Anwendersicht dauerhaften Betrieb ermöglicht. Praktisch wird

dieser allerdings durch Verschleiß (insbesondere die Zyklenfestigkeit bei Akkumulatoren) und die Verfügbarkeit der Energiequelle beschränkt.

Tabelle 2.2 stellt verschiedene Energiequellen dar, aus denen ein System versorgt werden kann. Die Menge, die gewonnen wird, hängt sehr stark von den Umgebungsparametern ab. Solarzellen als bekannteste Quelle liefern zwar unter Idealbedingungen die höchste Energie. Bei bedeckter Wetterlage oder wenn die Module abgeschattet sind, ist die Energieausbeute dagegen gering. Zusätzlich ist die Zeit, in der Energie gewonnen werden kann, vom Tag-Nacht-Zyklus abhängig. Im Gegensatz dazu können Technologien, die aus Funkwellen oder dem Wärmeunterschied Energie gewinnen, ohne Einschränkungen genutzt werden.

Tabelle 2.2: Aus der Umwelt gewinnbare Energie

Energiequelle	Gewinnbare Energie <sup>1</sup>	Quelle
Solarzelle - direktes Sonnenlicht	15 mW/cm <sup>2</sup>	[40]
Solarzelle - bewölkt	0,15 mW/cm <sup>2</sup>	[40]
Solarzelle - Innenraum	0,006 mW/cm <sup>2</sup>	[40]
Solarzelle - Tischlampe < 60W	0,57 mW/cm <sup>2</sup>	[40]
Piezoelektrisch - Schuheinlagen	0,33 mW/cm <sup>2</sup>	[40]
Vibration - Mikrowelle	0,01-0,1mW/cm <sup>2</sup>	[40]
Thermoelektrisch - 10°C Gradient	0,04 mW/cm <sup>2</sup>	[40]
Thermoelektrisch - Mensch	0,025 mW/cm <sup>2</sup>	[41]
Thermoelektrisch - Industriell	1-10 mW/cm <sup>2</sup>	[41]
Funkwellen - GSM	0,1 µW/cm <sup>2</sup>	[41]
Funkwellen - Wlan	0,001 µW/cm <sup>2</sup>	[41]
Vibration - menschliche Bewegung (Hz)	4 µW/cm <sup>3</sup>	[42]
Vibration - maschinelle Bewegung (kHz)	0,8 mW/cm <sup>3</sup>	[42]

<sup>1</sup> Die gewinnbare Energie hängt stark von der eingesetzten Technologie und den Umgebungsparametern ab.

## Systemaufbau

Der grundsätzliche Aufbau eines Systems zur Energiegewinnung hängt von der genutzten Energiequelle und den Anforderungen der Anwendung ab. Wenn die Energiequelle permanent verfügbar ist und der Bedarf direkt gedeckt werden kann, benötigen solche Systeme potenziell keinen Speicher. So wird in [43] die Energie für eine Übertragung durch einen Tastendruck erzeugt. Muss ein höherer pulsformiger Bedarf befriedigt werden, der nicht direkt zur Verfügung steht, muss die Energie gesammelt und gespeichert werden.

Der potenzielle Aufbau ist in Abbildung 2.9 dargestellt. Je nach verwendeter Hardware sind Spannungswandler notwendig, die eine für das eingebettete System passende Spannung erzeugen. Wenn Energie zwischengespeichert werden soll, kann dies mit einem oder zwei Speichern erfolgen. Zwar gibt es in der Literatur Beispiele,

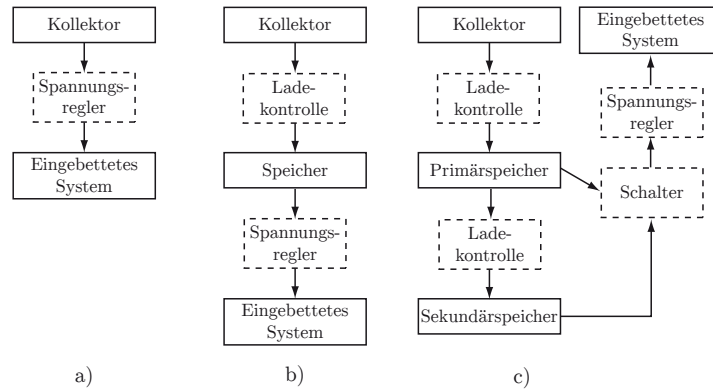


Abbildung 2.9: Aufbau eines Systems mit Energiegewinnung aus der Umwelt a) ohne Speicher b) mit einem Speicher c) mit zwei Speichern. (nach [44] verfeinert)

die als Primärspeicher Akkumulatoren verwenden [45]. Dies ist jedoch im Hinblick auf die beschränkte Zyklenfestigkeit problematisch. Systeme wie beispielsweise [46], die Superkondensatoren als Speicher nutzen, sind aufgrund der hohen Selbstentladung auf eine regelmäßige Energiezufuhr angewiesen. Wenn diese nicht vorausgesetzt werden kann, bietet sich ein hierarchischer Aufbau an, wie in [47] vorgestellt und Abbildung 2.9c skizziert. Als Primärspeicher wird ein Kondensator eingesetzt. Wenn dieser geladen ist, wird die gesammelte Energie zur Ladung des als Sekundärspeicher eingesetzten Akkumulators verwendet. Das eingebettete System kann zwischen beiden Energiequellen wählen, sodass bei einer vollständigen Entladung des Primärspeichers auf den deutlich größeren Sekundärspeicher ausgewichen werden kann. Gleichzeitig sorgt dieser Aufbau dafür, dass die Ladung des Sekundärspeichers besser auf Zyklenfestigkeit ausgerichtet werden kann.

### Energieneutralität

Oberstes Ziel eines drahtlosen tief eingebetteten Systems, das Energie aus der Umwelt sammelt, ist das Erreichen der sogenannten Energieneutralität [48]. Dabei soll nur soviel Energie verwendet werden, wie aus der Umwelt gesammelt werden kann. Auf der Ebene einzelner Systeme bedeutet dies theoretisch einen dauerhaften Betrieb. Im Falle eines Sensornetzes kann dies aber auch im Kontext des gesamten Netzes und dessen Aufgaben betrachtet werden, in dem einzelne Knoten für das Ausbringungsziel durchaus mehr Energie verbrauchen können, als sie selbst gesammelt haben. Steht die verwendete Energiequelle nicht permanent zur Verfügung und/oder ist in der Ausbeute nicht statisch, ist es notwendig, diese vorherzusagen. Dies kann auf verschiedene Arten erfolgen, wie beispielsweise in [22, 48, 49] gezeigt, setzt aber neben Informationen über vergangene Erträge immer Wissen über den Verbrauch des Systems und im Allgemeinen komplexe Berechnungen voraus, die ihrerseits Energie kosten.

### 2.1.4 Spannungsregler

Wird ein eingebettetes System direkt mit der Energiequelle verbunden, ist es der mit der Zeit sinkenden Spannung der Quelle ausgesetzt. Dies kann den Verbrauch des Systems beeinflussen, da Spannungsabhängigkeiten im Energieverbrauch und in der Leistung integrierter Systeme bestehen [34]. Des Weiteren kann die Versorgungsspannung unter einen für das eingebettete System oder dessen Teile verwendbaren Wert fallen und damit das System ausfallen. Auch kann der Spannungsbereich der Energiequelle ganz außerhalb des vom angeschlossenen System Verwertbaren liegen.

In solchen Fällen muss durch die Nutzung eines Spannungsreglers die Spannung erhöht oder verringert werden. Diesem Zweck dienen sogenannte Aufwärtsregler (engl. step-up-(Boost)-converter) für die Erzeugung einer höheren Spannung und Abwärtsregler (engl. step-down-(Buck)-converter) für eine niedrigere Spannung. Diese können verschieden aufgebaut sein. Allen gemein ist aber, dass für die Wandlung selbst Energie verbraucht wird.

In [50] wird experimentell auf die Kapazität von Lithium Ionen Knopfzellen im Zusammenhang mit Spannungsreglern eingegangen. Es zeigte sich, dass die Energiemenge, die von der Schaltung verbraucht wird, signifikant von der der Batterie entnommenen abweicht. Dies ist auf den Verlust durch den Regler zurückzuführen, da dieser eine Effizienz von nur rund 70% aufweist.

In [51] wird ein Experiment beschrieben, das im Jahr 2002 durchgeführt wurde. Aufgrund ihrer Erfahrungen raten die Autoren von der Benutzung von Boost-Konvertern ab, da diese mit 15% zusätzlicher Leistung weit hinter den Erwartungen zurückgeblieben sind und deshalb nicht wirtschaftlich seien.

Die teils geringe Effizienz ist besonders dann problematisch, wenn sehr wenig Energie verbraucht wird, z. B. wenn sich das eingebettete System im Schlafmodus befindet. Zusätzlich hängt die Effizienz von der Eingangsspannung und dem aktuellen Verbrauch des angeschlossenen Systems ab. Zu der wechselnden Effizienz kommt der Eigenverbrauch eines Reglers, der den Verbrauch eines sehr sparsamen und selten aktiven drahtlosen tief eingebetteten Systems übersteigen kann.

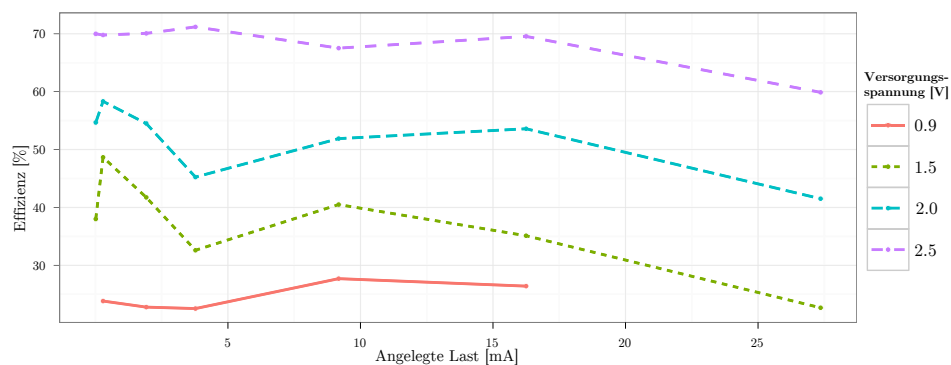


Abbildung 2.10: Gemessene Effizienz eines Aufwärtsregulators NCP1400A33T1



Die Effizienz für verschiedene Spannungs- und Lastbereiche eines NCP1400A33T1 Aufwärtsregulators [52] wurde durch Experimente ermittelt und ist in Abbildung 2.10 dargestellt. Je geringer die Versorgungsspannung ist, desto niedriger die Effizienz. Diese ist auch für niedrige Lasten geringer als für höhere.

Die inkonsistente und teils geringe Effizienz, der Eigenverbrauch und die Hardwarekosten, können die Vorteile eines Reglers nivellieren. Der Einsatz sollte vom Systemdesigner vorsichtig bedacht werden [34].

## 2.2 Komponenten und ihre Benutzung

Die Hardware eines drahtlosen tief eingebetteten Systems besteht aus einem Prozessor, einem Speicher, einem Kommunikationsmodul und einem oder mehreren Sensoren und/oder Aktuatoren (vergleiche Abbildung 2.1, Seite 9).

Für jede dieser Komponenten existieren verschiedene Ansätze, um den Energieverbrauch zu senken. Ein Ansatz, der für alle gleichermaßen möglich ist, aber von der Anwendung abhängt, ist die Nutzung eines Arbeitszyklus, des sogenannten *Duty Cycle*. Dabei iteriert ein Gerät, wie in Abbildung 2.11 dargestellt, zwischen einer aktiven und einer passiven Phase. Während in der aktiven Phase Werte gemessen, verarbeitet und kommuniziert werden, ist der Verbrauch in der passiven Phase minimal, das System ist größtenteils inaktiv und schläft. Je nach Dauer der jeweiligen Phase wird entsprechend Energie eingespart, um das Erreichen von hohen Standzeiten zu ermöglichen. Eine solche Einteilung hat jedoch individuelle Auswirkungen auf die zu erwartende geleistete Arbeit der Komponenten und erfordert zum Teil Maßnahmen, um dies zu koordinieren. Zusätzlich müssen die Anforderungen an die Anwendung beachtet werden.

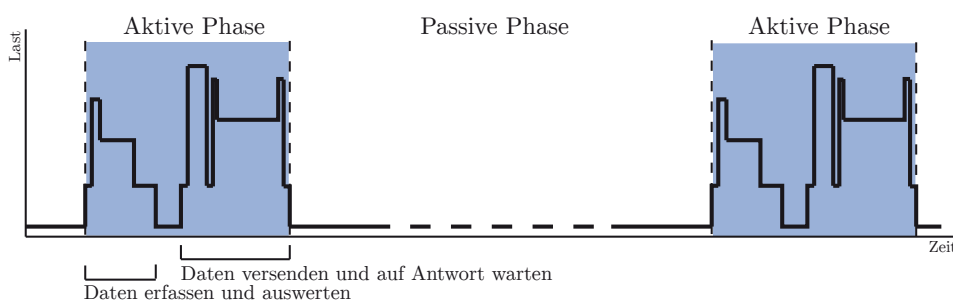


Abbildung 2.11: *Duty Cycling*: zeitliche Aufteilung der Aktivität in aktive und passive Phasen

### 2.2.1 Prozessor

Das Herz eines drahtlosen eingebetteten Systems ist der Prozessor. Dieser ist typischerweise in Form eines Mikrocontrollers ( $\mu\text{C}$ ) vorzufinden, der als System-on-

a-Chip (SOC) Prozessor, Speicher und Peripherie vereint. Die Peripherie umfasst verschiedene Timer, Bussysteme und Analog-Digital-Wandler (engl. analog digital converter - ADC). Zusätzlich können auch Echtzeituhren und Funkmodule integriert sein.

## Hardware

Es existiert eine Vielzahl verschiedener  $\mu\text{C}$  mit unterschiedlichsten Architekturen. Tabelle 2.3 stellt drei Vertreter mit unterschiedlichen Charakteristiken einander gegenüber. Sie unterscheiden sich nicht nur in der Architektur, sondern auch in der Leistungsfähigkeit und insbesondere dem Energieverbrauch. Darüber hinaus existieren die verschiedenen  $\mu\text{C}$ -Serien in unterschiedlichen Varianten, die sich in der vorhandenen Peripherie und Pin-Zahl unterscheiden.

Tabelle 2.3: Vergleich der Architekturen und Leistungsparameter verschiedener Mikrocontroller

	Atmel ATxmega128D4 [53]	Texas Instruments <sup>1</sup> CC430F6138 [54]	Energy Micro EFM32G890 [55]
CPU	AVR	MSP430	ARM Cortex-M3
Architektur	Harvard	Von-Neumann	Von-Neumann
Daten-/Adressbreite	8/16Bit	16Bit	32Bit
Spannungsbereich	1,6-3,6V	1,8-3,6V	1,8-3,8V
Takt (maximal)	32Mhz	20Mhz	32Mhz
Aufwachzeit	4,5 $\mu\text{s}$	<6 $\mu\text{s}$	2 $\mu\text{s}$
Aktiver Modus <sup>2</sup>	0,306mA/MHz	0,21mA/MHz	0,18mA/MHz
Schlafmodus <sup>3</sup>	17,3 $\mu\text{A}$	2,2 $\mu\text{A}$	0,9 $\mu\text{A}$
	1,3 $\mu$ ohne Timer		
ADC & Referenz	1,7mA	0,25mA	0,411mA
Flash Löschen	4mA	5mA	7mA
Flash Schreiben	4mA	2mA	7mA

<sup>1</sup> alle Werte für Powermode 2

<sup>2</sup> Befehlsausführung aus Flashspeicher, ermittelt aus Verbrauch bei maximalem Takt

<sup>3</sup> tiefstmöglicher Energiesparmodus mit RAM-Retention, 32kHz Clock für Timer, Watchdog und RTC aktiv

Es gibt eine Vielzahl unterschiedlicher Hersteller, die ARM-Kerne mit eigener Peripherie anbieten, sodass der dargestellte Controller nur als Beispiel für einen besonders energiesparenden Vertreter dienen kann. Andere ARM-basierte Controller sind nicht explizit auf Schlafmodi ausgelegt, sodass der tiefste mögliche Schlafmodus mit RAM-Erhaltung vergleichsweise viel Energie benötigt. So verbraucht der STM32F20xx [56] von STMicroelectronics im sogenannten STOP Modus 0,55mA. Tiefere Modi sind zwar verfügbar, allerdings ist in diesen nur ein kleiner Teil des RAMs mit Energie versorgt, sodass für die meisten Anwendungen eine aufwendige Logik zur Sicherung und Wiederherstellung des Programmzustandes implementiert

werden muss. Bei allen  $\mu\text{C}$  existieren tiefere Schlafmodi, welche aber weite Teile des Taktsystems deaktivieren und so die eventuell nötige Peripherie nicht mehr arbeiten lassen.

### Betriebssysteme

Mit seinem Programm steuert der Prozessor des  $\mu\text{C}$  das Verhalten aller anderen Komponenten. Dies kann sehr komplex werden und Abhängigkeiten können einen Wechsel der Hardwareplattform erschweren. Aus diesem Grund wurden für diese Kleinstsysteme spezielle Betriebssysteme entwickelt.

Betriebssysteme für tief eingebettete Systeme vereinfachen die Softwareentwicklung, denn sie helfen bei der Nutzung der Hardware durch Bereitstellen von standardisierten Abstraktionen und Schnittstellen und unterstützen damit die Portabilität. Ihr Ziel ist es, den Entwickler, neben Abhängigkeiten von der Hardware, von Aufgaben wie Synchronisation oder Scheduling zu befreien.

Grob lassen sich Betriebssysteme in zwei Kategorien oder Paradigmen einteilen: einerseits Thread-basierte Systeme, wie freeRTOS [57], Mantis [58] sowie Embedded-Unix/Linux oder Windows CE für leistungsfähigere Systeme und andererseits ereignisbasierte Systeme, beispielsweise SOS [59], TinyOS [60], Contiki [61] oder REFLEX [62]. Beide Paradigmen haben unterschiedliche Vor- und Nachteile (eine Diskussion findet sich in [63]).

Abbildung 2.12a) stellt die Zustandsautomaten einer vereinfachten Sensornetzwerdung dar, bei der Daten periodisch mit einem Sensor erfasst und per Funk versendet werden und Nachrichten anderer Knoten zu einer Senke weitergeleitet werden müssen. Wie in Abbildung 2.12b) und c) dargestellt ist, unterscheidet sich die Umsetzung dieser Anwendung je nach gewähltem Programmierparadigma. In ereignisbasierten Systemen werden Ereignisse nach ihrer Erzeugung zwischen voneinander unabhängigen bearbeitenden Komponenten weitergereicht. Diese Komponenten besitzen eine *run-to-completion*-Semantik, dürfen also nicht blockieren, da es nur einen einzelnen Kontrollfluss gibt. Im Gegensatz dazu können bei Thread-basierten Systemen mehrere parallele Kontrollflüsse existieren. Dort werden, wie in der Abbildung dargestellt, Operationen nacheinander ausgeführt. Muss auf die Fertigstellung einer Operation gewartet werden, blockiert ein Thread, wobei es für den Entwickler nicht immer ersichtlich ist, ob ein Funktionsaufruf zu einer Blockierung führt. Während Threads damit eher den von Entwicklern gewohnten Stil und Ablauf ermöglichen, entsprechen Ereignisse eher der Abarbeitung von Zustandsautomaten. Ereignisbasierte Systeme können jedoch zumindest dem Verhalten nach Threads durch Erweiterungen wie Protothreads [64] oder Tiny-Threads [65] nachbilden.

Das verwendete Programmierparadigma hat Auswirkungen auf die Möglichkeiten des Energiemanagements. Energiemanagement kann implizit erfolgen oder explizit

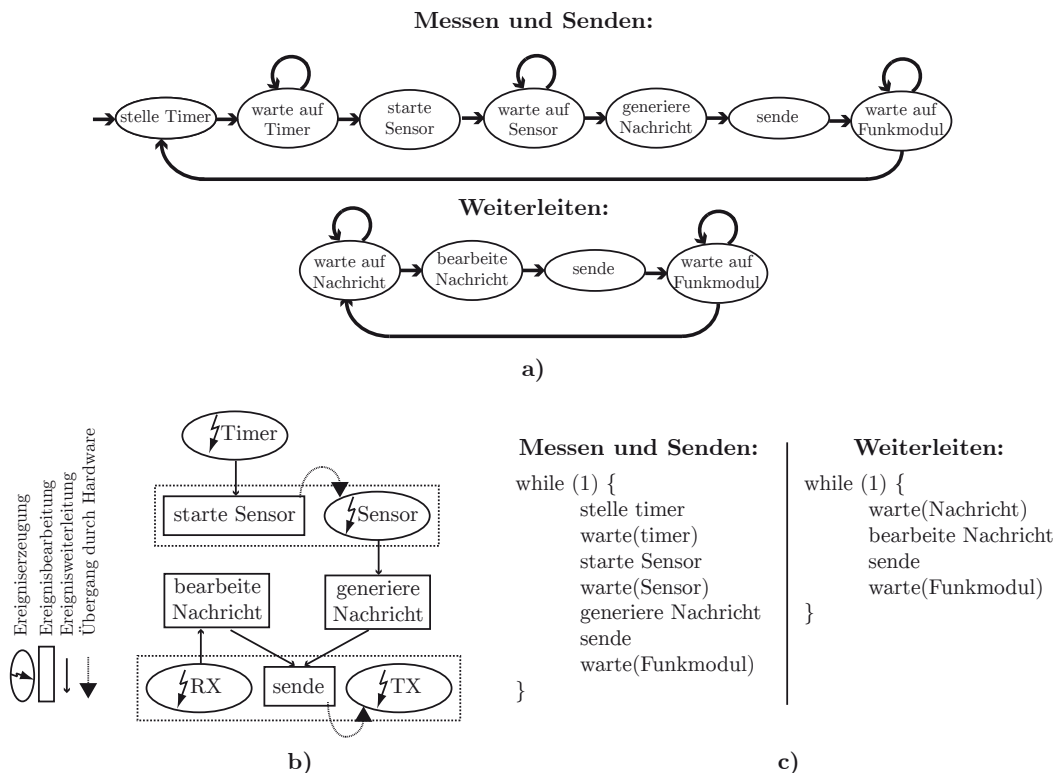


Abbildung 2.12: Darstellung einer vereinfachten Sensornetzanwendung mit zwei parallelen Ausführungspfaden als Automat a). Mögliche Umsetzungen als schematische Darstellung mittels Ereignissen b) und Pseudocode mittels Threads c)

durchgeführt werden. Implizites Management umfasst die Möglichkeit, Hardware auf Anfragebasis zu aktivieren oder zu deaktivieren, ohne dass das Anwendungsprogramm eingreifen muss, zum Beispiel beim Versenden von Funknachrichten. Geräte, die asynchron Daten liefern, beispielsweise das Funkmodul für das Empfangen von Nachrichten, müssen dagegen explizit aktiviert werden. Des Weiteren lässt sich der Verbrauch eines Systems explizit durch Veränderung von Systemparametern steuern.

Während diese Mechanismen sowohl bei Ereignis- als auch bei Thread-basierten Systemen identisch gelöst werden können, hat das jeweilige Programmierparadigma Auswirkungen auf die Möglichkeit, Totzeiten zu erkennen und potenziell durch einen Schlafzustand Energie zu sparen. In Thread-basierten Systemen muss der Scheduler ermitteln, ob alle Threads blockiert (z. B. weil sie auf Ein-/Ausgabe warten) oder idle sind. Ereignisbasierte Systeme sind in der Lage, nicht nur Zustandsübergänge der Software, sondern auch die der Hardware abzubilden. So kann die Zeit zwischen Übergängen gleichbedeutend mit Schlafzuständen der Hardware sein. Im Beispiel in Abbildung 2.12b) kann nach dem Starten des Sensors der Prozessor in einen Schlafmodus überführt werden, bis das Ende des Messvorgangs durch die Unterbrechung des Sensors angezeigt wird. Der Scheduler kann potenziell immer dann einen Wech-

sel des Prozessors in einen Schlafzustand anweisen, wenn keine Ereignisse anliegen und die Ereigniswarteschlange leer ist.

Ein weiterer Unterschied der Paradigmen liegt im softwaretechnischen Aufwand, der betrieben werden muss, um eine Anwendung mit Energiemanagementfunktionalität auszustatten. Bei ereignisbasierten Systemen ist dies durch Annotation der Ereignisse möglich und kann, sofern nur die Anwendungsstruktur beschreibender Quellcode betroffen ist, für die Anwendungskomponenten transparent sein. Durch die bereits erfolgte Dekomposition in Einzelkomponenten ist eine feingranulare Erfassung und Steuerung des Verbrauchs möglich. Dies lässt sich bei Thread-basierten Systemen nur durch eine Modifikation des Anwendungscodes, durch Annotation bzw. Aufspalten in mehr Threads erreichen.

Unabhängig vom verwendeten Programmierparadigma existieren Betriebssysteme und Erweiterungen bestehender Systeme, die der Ressource Energie eine besondere Bedeutung beimessen. Hierzu zählen beispielsweise Energy Levels [21], EON [66], EPOS [67], Nemesis OS [68] und ECOSystem [69]. Diese und weitere Vertreter und ihre Mechanismen werden in Kapitel 5.2 vorgestellt.

### Energiesparmaßnahmen

Moderne Mikroelektronik wird aktuell in CMOS-Technik (Complementary Metal Oxide Semiconductor) hergestellt. Die Leistungsaufnahme einer solchen CMOS-Schaltung besteht aus einem statischen  $P_{stat}$  und einem dynamischen  $P_{dyn}$  Anteil. Formel 2.2 beschreibt die Bestandteile des dynamischen Teils. In diesen fließen der Grad der Aktivität  $\alpha$ , die kapazitive Last  $C$ , die Spannung  $V$  und die Taktfrequenz  $f$  ein [70, S. 455].

$$P_{dyn} = \alpha * C * V^2 * f \quad (2.2)$$

Es existieren zwei allgemeine Ansätze, um den Energieverbrauch einer solchen Schaltung zu senken. Zum einen die dynamische Spannungs- und Frequenzanpassung (engl. dynamic voltage and frequency scaling - DVFS). Zum anderen die Verringerung des Aktivitätsgrades, was technologisch als Abschaltung des Taktes (engl. clock gating) und/oder der Versorgungsspannung (engl. power gating) von Teilen des SOC umgesetzt wird und bei  $\mu C$  in Form von diversen Schlafmodi genutzt werden kann. Letzteres entspricht dem am Anfang von Kapitel 2.2 vorgestellten *Duty Cycling*.

**Dynamische Spannungs- und Frequenzanpassung** Zwar sind alle aktuellen Mikrocontroller in der Lage, ihren Takt softwaregesteuert zu verändern, jedoch wirkt sich dies auf die Ausführungsgeschwindigkeit aus. Eine Halbierung des Taktes halbiert zwar den Stromverbrauch, verdoppelt aber ebenso die Abarbeitungsdauer.

Wirkliche Sparpotentiale ergeben sich aus der Verringerung der Spannung, da diese einen quadratischen Einfluss auf den Verbrauch hat. Allerdings kann die Spannung nicht beliebig verringert werden und wirkt sich auch auf den erreichbaren Takt aus. Die Kombination aus beidem, DVFS, bietet die besten Sparpotenziale, da die Frequenz entsprechend den zeitlichen Applikationsanforderungen eingestellt werden kann, woraufhin die Spannung auf den passenden Wert geregelt wird.

Diese Technik wird allerdings kaum bei tief eingebetteten Systemen genutzt. Das liegt unter anderem daran, dass bei geringer Auslastung die Kosten der Spannungsanpassung die Einsparungen überwiegen [71]. In [72] wird eine Variante für tief eingebettete Systeme vorgestellt, das sogenannte *passive voltage scaling*. Dabei wird auf einen Spannungsregler verzichtet und das System direkt an die Batterie angeschlossen. Der Takt wird dann an die verfügbare Spannung angepasst und muss nicht statisch auf die niedrigste Spannung eingestellt werden. Durch den Verzicht auf einen Wandler und auf einen statischen Takt kann das System von einer höheren Leistung in der Anfangsphase und einer verbesserten Laufzeit profitieren. Der verringerten Leistung durch Reduzierung des Taktes mit sinkender Spannung wird in [73] durch Aktivierung zusätzlicher Kerne eines eingebetteten Multicores begegnet.

Eine Kombination aus DVFS und PVS wird in [74] vorgestellt. Um die bestmögliche Kombination aus Versorgungsspannung und Frequenz erreichen zu können, wird ein System aus mehreren Spannungsreglern mit unterschiedlichen Charakteristiken und der direkten Versorgung verwendet. Je nach Anforderungen an die Frequenz wird dynamisch die effizienteste Art der Versorgung gewählt und somit der Energieverlust durch Wandlungskosten verringert.

**Schlafmodi** Ein bei  $\mu\text{C}$  häufig genutzter Weg, um den Energieverbrauch zu reduzieren, ist die Überführung des Prozessors in einen Schlafzustand. Je nach Tiefe des Zustandes sind verschiedene Teile des SOC deaktiviert und die Möglichkeiten, den Prozessor wieder aufzuwecken, geringer. Um die Schlafmodi nutzen zu können, muss der Prozessor wieder geweckt werden können. Dies hängt auch von der Art und Weise ab, wie dieser mit der Peripherie kommuniziert. Zum einen existiert die Möglichkeit, Geräte zyklisch abzufragen (engl. *polling*), dies verhindert allerdings, dass der Prozessor in einen Schlafzustand wechseln kann. Zum anderen können Unterbrechungen genutzt werden, bei denen der Prozessor sowohl von externen Quellen (Sensoren und Kommunikation) als auch von internen Quellen (Timer und Echtzeituhr) aus geweckt werden kann.

Da Mikrocontroller über eine Vielzahl von Peripherie und Schlafmodi verfügen, ist die Wahl des passenden Modus nicht trivial. Während ein zu hoher Modus letztlich nur Energie verschwendet, verhindert ein zu tiefer Modus im schlimmsten Fall, dass das System wieder aufgeweckt wird. Ein Betriebssystem sollte deshalb die Wahl zumindest unterstützen oder implizit treffen. Existierende Betriebssysteme lösen dies

unterschiedlich. Zum Teil werden Schlafmodi nicht unterstützt [57, 59] und müssen vom Nutzer implementiert werden. Andere Systeme [58, 61] unterstützen Schlafmodi und stellen Funktionen für deren Nutzung bereit. Manche Systeme bieten eine implizite Ermittlung des tiefst möglichen Schlafmodus [75, 76, 77].

### 2.2.2 Kommunikation

Kommunikation ist für drahtlose tief eingebettete System essentiell. Sie ermöglicht die Erfüllung ihrer Aufgaben und die Kooperation miteinander. Kabelbasierte Systeme sind oftmals unflexibel und verursachen einen hohen Ressourcenaufwand bei der Installation, weshalb sich die Nutzung von Funk anbietet. Allerdings ist die Zuverlässigkeit gegenüber drahtgebundener Kommunikation deutlich geringer, was zu einem erhöhten Aufwand für die Kommunikation selbst führt. Je nach verwendetem Zugriffsverfahren wird das Medium von allen Teilnehmern geteilt. Die Störreichweite liegt weit über der eigentlichen Sendereichweite. Die überwiegende Mehrheit der Verbindungen ist unidirektional und die Änderungsrate der Verbindungen ist sehr hoch [78]. Zusätzlich kann es zum *hidden-station*-Problem und zum *exposed-station*-Problem kommen [79, S. 70]. Um trotzdem erfolgreich Daten zu übertragen, existiert eine Vielzahl an Medienzugriffsprotokollen (engl.: media access control - MAC) mit unterschiedlichen Zielsetzungen und Ansätzen. Wenn eine direkte Verbindung nicht möglich ist, müssen Wegewahlverfahren (engl. routing) eingesetzt werden, um die Nachrichten zuzustellen. Insbesondere im Bereich der drahtlosen Sensornetze existieren viele verschiedene Ansätze, oftmals ebenenübergreifend in Kombination mit einem MAC. Hinzu kommt, dass die Energiekosten funkbasierter Kommunikation deutlich höher sind als die Kosten von Berechnungen. Abhängig von der verwendeten Hardware können 1000 Bytes berechnet oder 1 Byte übertragen werden [80]. Diese Kosten müssen im Falle von Mehr-Hop-Verbindungen von allen getragen werden, nicht nur von Sender und Empfänger. Eine direkte Kommunikation ist deshalb, trotz des möglicherweise höheren individuellen Verbrauchs, global gesehen günstiger [81].

### Hardware

Mehrere Bereiche des Frequenzspektrums stehen der Kommunikation zur Verfügung. Im Bereich der drahtlosen tief eingebetteten Systeme wird üblicherweise das ISM-Band genutzt. Tabelle 2.4 gibt einen Überblick über Funkmodule in Frequenzbereichen, die in Europa nutzbar sind. Gerade 2,4 GHz wird intensiv genutzt, was die Fehlerwahrscheinlichkeit durch Kollisionen von Nachrichten erhöht und zum Beispiel bei Bluetooth [87] durch Nutzung von Frequenzsprungverfahren [79, S. 211] kompensiert wird.

Tabelle 2.4: Vergleich Architekturen und Leistungsparameter verschiedener Funkmodule

Chip	2,4GHz				433MHz	868MHz
	CC2520 [82]	CC2570 [83]	nRF8001 [84] <sup>1</sup>	CC3000 [85] <sup>2</sup>		
Standard	IEEE 802.15.4	-	IEEE 802.15.1	IEEE-802.11abg	-	-
Protokoll	Zigbee	ANT / ANT +	Bluetooth low power	WLAN	-	-
Spannungsbereich	1,8-3,8V	2,0-3,6V	1,9-3,6V	2,7-4,8V	1,8-3,6V	1,8-3,6V
max Datenrate	250kbps	1Mbps	1Mbps	54 Mbps	600kbps	600kbps
Durchschnittliche Stromaufnahme bei 3,0V						
Sleep	0,175µA	0,5µA	0,5µA	<5µA	0,2µA	16,9mA
RX	18,8 mA	23,7 mA	14,6mA	92mA	17,1mA	34,2mA/12dBm
TX max / dBm	33,6mA/5dBm	34,3mA/4dBm	12,7 mA/0dBm	190mA/14dBm	29,2mA/10dBm	16,4mA/-6dBm
TX min / dBm	25,8mA/0dBm	25,9mA/-6dBm	8,4mA/-18dBm	-	13,1mA/-6dBm	16,4mA/-6dBm
µJ/Bit max/min	0,4/0,3	0,1/0,077	0,038/0,025	0,012/-	0,17/0,078	0,2/0,98

<sup>1</sup> Verbrauchswerte bei deaktiviertem Spannungsregler

<sup>2</sup> Verbrauchswerte bei 3,6V

<sup>3</sup> µJ/Bit auf Basis von 500kbps



Die Module unterscheiden sich in ihren Charakteristika, wobei die Datenrate und die Stromaufnahme für die Auswahl einer Technologie letztlich die größte Rolle spielen. Zwar hat das WLAN-Modul den höchsten Verbrauch, dieser relativiert sich jedoch, wenn man die Datenrate einbezieht. Verglichen damit ist die Übertragung eines einzelnen Bits mit dem Zigbee Modul, welches absolut eine fünf mal kleinere Stromaufnahme besitzt, durchschnittlich 33 mal so energieintensiv.

Die Sendekosten allein sind jedoch größtenteils von untergeordneter Bedeutung. Schwerwiegender ist für die meisten Anwendungsszenarien die Stromaufnahme im Empfangsmodus. Um eine Nachricht empfangen zu können, ist es notwendig, den Kanal abzutasten. Da dies aber nicht immer nur dann erfolgt, wenn eine Nachricht auch tatsächlich empfangen wird, findet oft ein sogenanntes *idle listening* statt, bei welchem viel Energie verschwendet wird. Um dies zu reduzieren, wird *Duty Cycling* eingesetzt und die Module werden periodisch in einen energiearmen Schlafzustand versetzt. Allerdings erhöht dies den Aufwand für die Synchronisation der Kommunikationspartner. Die Umsetzung kann im MAC-Protokoll oder unabhängig davon erfolgen.

Eine alternative Lösung wäre der Einsatz von Wakeup-Funkmodulen. Sie basieren auf der Idee, dass ein zweites Kommunikationsmodul auf einem Sensorknoten vorhanden ist, das sehr wenig [88, 89], im Idealfall sogar gar keine [90] Energie verbraucht, während auf Nachrichten gelauscht wird. Es nutzt die Energie auf dem Medium zum Aufwecken eines Knotens, wenn Nachrichten für diesen anliegen. Das *idle listening* wird somit komplett eliminiert oder zumindest nur wenig Energie verbrauchen. Bisher sind solche Module allerdings nur prototypisch und nicht kommerziell verfügbar.

### Medienzugriffsprotokolle

Zur Koordination der Kommunikation werden Medienzugriffsprotokolle und Auswahlverfahren oder eine Kombination aus beidem eingesetzt. Die einzelnen Medienzugriffsprotokolle lassen sich grob in planbasierte Verfahren und konkurrenzbasierte Verfahren einteilen, wobei Mischverfahren ebenso existieren:

**Planbasierte Verfahren** koordinieren die Kommunikation anhand eines vorher festgelegten Plans. Dieser Plan legt fest, wer senden kann, und zum Teil auch, wer hören muss. Eingesetzt werden Multiplexverfahren (engl. multiple access) auf Basis der Zeit (engl. Time Division Multiple Access - TDMA) [79, S. 73], des Codes (engl. Code Division Multiple Access - CDMA) [79, S. 82], des Ortes (engl. Space Division Multiple Access - SDMA) [79, S. 72] oder der Frequenz (engl. Frequency Division Multiple Access - FDMA) [79, S. 72]. Zum Teil werden auch Mischverfah-

ren eingesetzt. Beispielsweise wird in einem weitläufigen Sensornetz implizit SDMA verwendet, da eine räumliche Trennung von Teilbereichen existiert.

**Konkurrenzbasierte Verfahren** konkurrieren bei jeder Nachricht um das Medium. Dazu werden Verfahren eingesetzt, um Kollisionen zu vermeiden. Vielfach wird eine Trägerprüfung (engl. Carrier Sense Multiple Access - CSMA) [79, S. 76] genutzt. Während in drahtgebundenen Netzwerken Kollisionen erkannt (CSMA/CD) [79, S. 70] werden können, ist in drahtlosen Verbindungen nur die Vermeidung möglich (CSMA/CA) [79, S. 77]. Dabei wird vor dem Senden einer Nachricht gelauscht und nur dann gesendet, wenn das Medium frei ist. Ist das Medium belegt, wird eine zufällig bestimmte Zeit lang gewartet. Zusätzlich können Mechanismen für eine virtuelle Kanalprüfung eingesetzt werden, bei der das Medium mithilfe von RTS/CTS-Paketen (Request to Send / Clear to Send) für eine gewisse Zeit reserviert wird. Dabei wird das *hidden-station*-Problem gelöst, da Dritte diese Reservierung wahrnehmen.

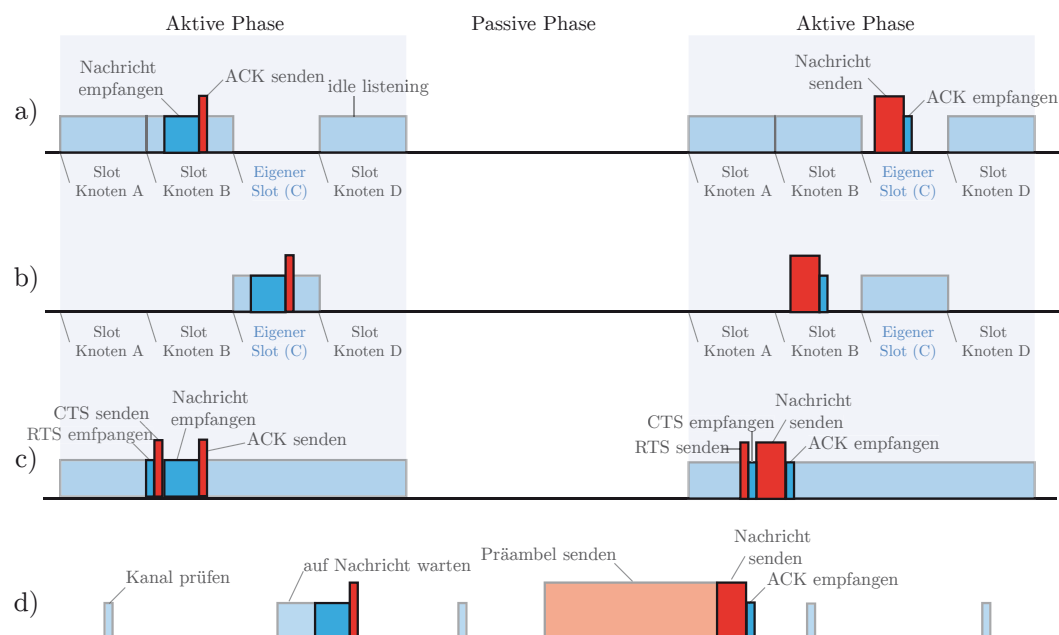


Abbildung 2.13: Vereinfachter Vergleich der Aktivität des Radiomoduls (blau - Empfangsmodus, rot - Sendemodus) für verschiedene MAC-Konzepte und *Duty Cycling*  
a) Sender-basiertes TDMA, b) Empfänger-basiertes TDMA, c) CSMA/CA mit virtueller Kanalprüfung, d) Low-power-Listening mit langer Präambel vor dem Senden

Das Nutzungsschema der Hardware und damit der Energieverbrauch unterscheiden sich von Protokoll zu Protokoll. Abbildung 2.13 vergleicht vereinfacht die Aktivität des Funkmoduls bei verschiedenen Ansätzen des Medienzugriffs. Die Abbildung verzichtet auf eine Darstellung der Initialisierungsphasen, beinhaltet aber die Ein-

teilung in aktive und passive Phasen. Innerhalb der ersten aktiven Phase wird ein Paket empfangen, innerhalb der zweiten ein Paket versendet. Der Empfangsmodus ist in der Abbildung blau dargestellt und der Sendemodus rot.

Die beiden ersten Ansätze folgen dem TDMA-Schema, unterscheiden sich aber in ihren Aktivitäten deutlich. Während a) Sender-basiert ist, also vorgegeben wird, wann ein Knoten senden darf, ist b) Empfänger-basiert und schreibt vor, wann ein Teilnehmer empfangsbereit ist. Die durch *idle listening* verbrauchte Energiemenge ist bei a) deutlich größer als bei b), allerdings wird dies durch einen potenziell höheren Aufwand erkauft, da jeder Knoten die Zeitschlitze seiner Nachbarn kennen muss. Zusätzlich sind Maßnahmen zur Kollisionsvermeidung notwendig.

Im konkurrenzbasierten CSMA/CA in Abbildung 2.13 c) existieren keine Zeitschlitze. Die Knoten können jederzeit innerhalb der aktiven Phase senden, sofern das Medium frei ist. Hierzu ist auch die Verwendung von RTS/CTS-Paketen illustriert.

Ganz ohne explizite aktive oder passive Phasen kommt das ebenfalls konkurrenz-basierte *low-power-listening* aus, bei dem periodisch kurz der Kanal geprüft wird [91]. Wenn diese Prüfung anzeigt, dass ein Sendevorgang vorliegt, bleibt der Knoten so lange empfangsbereit, bis eine Nachricht empfangen wurde oder ein Timeout erfolgt. Nachteil dieses Verfahrens sind die Sendekosten, da entweder vor jeder Nachricht eine ausreichend lange Präambel gesendet werden oder die Nachricht solange wiederholt werden muss, bis der Empfang bestätigt wird.

Die Aktivitäten und somit die nötige Energiemenge unterscheiden sich zwischen den Ansätzen a) bis c) deutlich, zum Teil ist der Verbrauch abhängig von der vorliegenden Situation. Letzteres macht die exakte Vorhersage des Verbrauchs schwierig, zumal viele Protokolle deutlich komplexer sind.

Tabelle 2.5 kategorisiert existierende MAC-Protokolle hinsichtlich eines variablen Energieverbrauchs (ohne Sendemodus). Einerseits kann sich die Dauer der Periode verändern, andererseits kann das Verhalten innerhalb einer Periode unterschiedlich sein. Ursache kann das (vermeintliche) Empfangen einer Nachricht sein oder eine Anpassung aufgrund von Umgebungsparametern, Energie und/oder Last. Zusätzlich ist es möglich, dass die Software des Knotens die Parameter des MACs explizit steuern kann, um sein Verhalten - ohne dabei die Konnektivität zu verlieren - mit dem Ziel anzupassen, gegebenenfalls Energie zu sparen oder Latenzen zu verringern. Allerdings ist die Mehrzahl der Protokolle auf Kapselung ausgelegt und arbeitet autonom. Da gerade im Sensornetzbereich sehr viele Protokolle existieren, wird kein Anspruch auf Vollständigkeit der Protokolle erhoben; die Tabelle ist exemplarisch.

Ein von den äußeren Umständen abhängiger Verbrauch macht eine Vorhersage, wie bereits beschrieben, schwierig. Hinzu kommt, dass ein Energiemanagement mit solchen Varianzen umgehen können muss. Daher ist eine von der Applikation und

Tabelle 2.5: Vergleich von Medienzugriffsprotokollen hinsichtlich variablen Energieverbrauch während der Laufzeit

		Interval/Periode	
		fix	dynamisch
Aktivität per Periode	fix	SMAC [92] LMAC [94] Zigbee WirelessHART [96]	AI-LMAC [93] DMAC [95]
	dynamisch	TMAC [97] B-MAC [91] Contiki-MAC [100] PMAC <sup>1</sup> [102] Bluetooth [103] TRAMA <sup>1</sup> [103]	ML-MAC-UL [98] ADAPT(Zigbee) [99] X-MAC [101]

<sup>1</sup> Variabilität explizit durch Software steuerbar

dem Management steuerbare Anpassung des MAC-Protokolls im Hinblick auf Lebenszeitziele wünschenswert.

### Wegwahlverfahren

Für eine Multihop-Kommunikation müssen Nachrichten von einem Knoten zum anderen weitergeleitet werden. Im einfachsten Fall leiten alle Knoten, die eine Nachricht empfangen, die nicht für sie bestimmt ist, diese einmal weiter. Die Nachricht wird gewissermaßen durch das Netz geflutet. Dieses Verfahren erzielt zwar eine hohe Auslieferungsrates, ist aber sehr ineffizient. Oftmals ist es besser, ein optimiertes Wegwahlverfahren einzusetzen. Ziel eines solchen Verfahrens ist es, Nachrichten sicher von der Quelle zum Ziel zu leiten und dabei unnötige Redundanzen zu vermeiden, um so den Energieverbrauch zu minimieren. Traditionell erhält jeder Knoten eine eindeutige Nummer, um eine Identifizierung und Adressierung zu ermöglichen und Pfade bilden zu können, jedoch weichen manche Protokolle von diesem Schema ab. In [104] werden Wegwahlverfahren in folgende Kategorien eingeteilt:

**Lokalitäts-basierte Protokolle** wie GAF [105] und GEAR [106] adressieren Knoten anhand ihrer Position. Da jeder Knoten seine Position kennt, können die Entfernung und folglich der Energieverbrauch eines Pfades ermittelt werden.

**Datenzentrische Protokolle** wie Directed Diffusion [107] und Rumor Routing [108] weichen vom traditionellen Adressierungsschema ab. Es werden nicht mehr einzelne Knoten angesprochen, sondern alle Knoten, die ein vorgebendes Kriterium

erfüllen. Zwischenstationen können Daten aggregieren und so das Nachrichtenaufkommen reduzieren.

**Hierarchische Protokolle** wie LEACH [109] und APTEEN [110] teilen ein Netz in disjunkte Cluster ein, innerhalb derer jeder Knoten einen dedizierten Ansprechpartner hat. Diese sogenannten Cluster Heads koordinieren die Kommunikation innerhalb des Clusters und sind für die Weiterleitung zwischen den Clustern verantwortlich. Die Wahl des Cluster Heads kann von verschiedenen Faktoren abhängen, beispielsweise von der verbleibenden Energiemenge.

**Mobilitäts-basierte Protokolle** wie Mobiroute [111] und Data MULES [112] setzen mobile Senken ein, um das Problem der Überbeanspruchung von senkennahen Knoten bei statischen Senken zu verringern. Die Mobilität bringt jedoch eine erhöhte Dynamik und einen größeren Aufwand bei der Auswahl des Weges mit sich.

**Mehrfad-basierte Protokolle** wie Braided Multipath [113] ermitteln nicht nur einen Pfad zum Ziel, sondern wählen zwischen verschiedenen Möglichkeiten, um die Last so besser zu verteilen.

**Heterogenitäts-basierte Protokolle** wie Energy-aware directed diffusion [80] und CHR [114] nutzen heterogene Knoten: zum einen die traditionellen batterieversorgten und deshalb eingeschränkten Knoten, zum anderen solche, die an eine Stromversorgung/Energiegewinnung angeschlossen sind. Letztere werden bei der Wegewahl priorisiert.

**QoS-basierte Protokolle** wie SPEED [115] und AntSenseNet [116] suchen die Balance zwischen Energieverbrauch und Qualitätsforderungen wie Verzögerung und Fehlertoleranz.

Alle Verfahren versuchen, auf unterschiedliche Arten das Problem der energetisch ausgetretenen Pfade zu lösen. Knoten, die viele Nachrichten weiterleiten müssen, verbrauchen potenziell mehr Energie als solche, die keine oder nur wenige Nachrichten anderer Knoten weiterleiten. Dies ist insbesondere dann problematisch, wenn eine oder mehrere explizite Senken in einem Netz existieren. Alle Knoten in unmittelbarer Nachbarschaft der Senke müssen die größte Last tragen und haben dementsprechend den potenziell höchsten Energieverbrauch. Dieses Gefälle im Energieverbrauch muss von einem Energiemanagement zumindest erfasst werden. Besser wäre es, wenn in Zusammenarbeit mit dem MAC-Protokoll und dem Wegewahlverfahren der Gesamtverbrauch des Knotens reguliert und somit ein verfrühter Ausfall verhindert werden könnte, da es sonst zu Netzwerkseparationen kommen kann.

### 2.2.3 Sensorik & Aktuatorik

Sensoren dienen der Erfassung von Messgrößen aus der Umwelt, wohingegen Aktuatoren die Umwelt manipulieren. Für letzteres muss elektrische Energie, soweit nicht direkt nutzbar, in Bewegung, Licht oder Schall umgewandelt werden. Beispiele hierfür sind Motoren für die Steuerung von Bewässerungsanlagen oder die Fenstersteuerung und Schalter/Dimmer für die Beleuchtung. Aber auch die Erzeugung von Stromstößen bei Herz-/Hirnschrittmachern oder zur Kontrolle von Tieren ist eine Manipulation der Umwelt durch Aktuatoren.

Der Fokus der drahtlosen tief eingebetteten Systeme liegt aber überwiegend auf der Datenerfassung. Daher beschränken sich die folgenden Ausführungen auf die Sensorik, insbesondere, da diese Optionen zur Einsparung von Energie eröffnet.

#### Hardware

Ein Sensor wandelt eine Messgröße in ein elektrisches Signal. Dieses kann analog sein oder bereits in digitaler Form vorliegen. Während analoge Signale zusätzlich einen Analog-Digital-Wandler benötigen, der wiederum selbst Energie verbraucht, können digitale Signale direkt verwendet werden. Bei solchen komplexeren und dementsprechend kostenintensiveren Sensoren sind meist der ADC und ein in Hardware implementiertes Busprotokoll integriert. Oftmals lassen sich solche Sensoren, im Gegensatz zu einfachen analogen Sensoren, umfangreich konfigurieren. So lässt sich beispielsweise die Rate der Datenerfassung oder die Auflösung des Sensors einstellen. Einige Sensoren verfügen zusätzlich über Alarm-Funktionalität, können also bei Erreichen einer vorher festgelegten Bedingung den angeschlossenen  $\mu\text{C}$  (per Interrupt) informieren. Durch die Nutzung standardisierter Busse ist es einfacher, mehrere Sensoren in ein System zu integrieren.

Für jede Messgröße existiert eine Vielzahl an unterschiedlichen Sensoren verschiedener Hersteller, die sich in ihren Details unterscheiden. Oft kann zwischen analogen und digitalen Sensoren gewählt werden, wobei letztere mit verschiedenen Busprotokollen verfügbar sind. Die Auswahlkriterien beschränken sich also nicht auf direkte Leistungsparameter und den Energieverbrauch, sondern die Wahl muss auch aufgrund der verfügbaren Anschlussmöglichkeiten innerhalb des Systems erfolgen. Tabelle 2.6 zeigt eine Auswahl an Sensoren für verschiedene Messgrößen unter Berücksichtigung des Strombedarfs. Dieser reicht von wenigen  $\mu\text{A}$  bis 100mA und mehr. Zum Teil ist der Verbrauch durch Anpassung der Betriebsmodi variabel. Nicht alle Sensoren verfügen über einen Schlafmodus, das heißt, sie müssen von der Spannungsversorgung getrennt werden, was zusätzliche Ressourcen im System notwendig macht.

Tabelle 2.6: Vergleich der Stromaufnahme von verfügbaren Sensoren für verschiedene Messgrößen

Sensor	Messgröße	Art	Stromaufnahme	
			Aktiv	Schlaf
GP2Y0A02YK[117]	Entfernung	Analog	33mA	3 $\mu$ A
MD62[118]	Co2 Gas	Analog	100mA	-
SFH5711[119]	Umgebungslicht	Analog	410 $\mu$ A	-
LIS3MDL[120]	3D Magnetfeld	Digital	270 $\mu$ A	1 $\mu$ A
NL-501ETTL[121]	Position	Digital	62mA	-
CMA3000-D01[122]	Beschleunigung	Digital	70/50/11/7 $\mu$ A @ 400/100/40/10 Hz	3 $\mu$ A
BMP085[123]	Barometrischer Druck	Digital	3-12 $\mu$ A @ 1Sample/s	0,1 $\mu$ A
FCH-m-PP-LC[124]	Durchfluss	Digital	25mA	-
SHT21[125]	Temperatur  Luftfeuchte	Digital	300 $\mu$ A - 66ms@14Bit, 8ms@11Bit 300 $\mu$ A - 22ms@12Bit, 3ms@8Bit	0,15 $\mu$ A
PIR-STD-LP[126]	Bewegung	Beides	40 $\mu$ A Ruhe, 400 $\mu$ A Bewegung	-

### Energiesparmaßnahmen

Der Verbrauch einiger Sensoren ist größer als der Bedarf von Funkmodulen (vergl. Tabellen 2.4 und 2.6). Um diesen Verbrauch zu senken, kann *Duty Cycling* angewendet werden. Allerdings kann dies in Abhängigkeit von der Messgröße zum Verlust von Ereignissen führen. Denn im Gegensatz zur Funkkommunikation kann keinerlei Synchronisierung erfolgen, verpasste Ereignisse werden auch nicht erneut übertragen und bleiben somit verloren. Dies macht die Entscheidung, ob ein Sensor ausgeschaltet werden kann, umso schwieriger. Hinzu kommt bei manchen Sensoren eine lange Startphase, bis Werte verfügbar sind. Wenn das Abtastintervall klein und die Startphase lang ist, wird die Nutzung von *Duty Cycling* erschwert oder verhindert. Über das einfache statische *Duty Cycling* hinaus gibt es Wege, den Energieverbrauch der Datenerfassung zu senken [127]:

**Adaptives Sampling** zielt darauf ab, nur dann Werte zu erfassen, wenn diese von Interesse sind [128, 129]. Dabei werden zeitliche oder räumliche Korrelationen der zu erfassenden Werte ausgenutzt. Wenn sich Werte innerhalb eines Zeitraumes und/oder einer Fläche nur geringfügig ändern, ist es unter Beachtung des Nyquist-



Shannon-Abtasttheorems [130] möglich, die Erfassungsrate zu verringern und so Energie zu sparen. Abbildung 2.14 stellt dies beispielhaft dar. Geringe Veränderungen der Messgröße sind uninteressant und bedürfen keiner hohen Erfassungsrate. Von Bedeutung sind große Veränderungen und Anomalien. Gerade letztere können allerdings unbemerkt bleiben, wenn die Erfassungsrate zu gering ist.

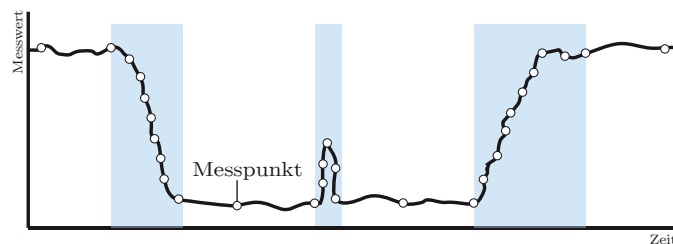


Abbildung 2.14: Adaptive Datenerfassung - Geringe Schwankungen benötigen kleinere Erfassungsraten als große Schwankungen und Anomalien

**Getriggerte/hierarchische und Multiskalendatenerfassung** verringern den Energieverbrauch durch Verwendung mehrerer Sensoren, die zwar dieselbe Messgröße erfassen, sich aber in der Genauigkeit und dem Energieverbrauch unterscheiden. So ist es möglich, genauere und dementsprechend energiehungrigere Sensoren nur dann zu verwenden, wenn die einfacheren Sensoren ein Ereignis feststellen, das mit höherer Auflösung erfasst werden soll. Wenn wie in [131] ein Knoten mehrere Sensoren besitzt, wird dies als getriggerte Datenerfassung bezeichnet. Es ist möglich, hierfür auch nur einen Sensor zu verwenden, wenn dieser verschiedene Auflösungen mit unterschiedlichen Energieprofilen bereitstellt. Bei der für Sensornetze geeigneten multiskalen Datenerfassung, beispielsweise [132], wird ein Bereich mit Sensoren überwacht. Wird ein Ereignis festgestellt, wird durch die Senke ein Roboter mit besserer Sensorik an diese Position geschickt.

**Modellbasiertes adaptives Sampling** basiert auf der Erstellung und Nutzung eines Vorhersagemodells, welches die zeitliche und/oder räumliche Korrelation lernt [133, 134]. Die Erfassungsrate kann somit verringert werden, da vorhergesagte Werte genutzt werden können. Weicht die Vorhersage zu sehr von einem tatsächlich erfassten Wert ab, ist das System in der Lage, das Modell anzupassen.

Eine ähnliche, allerdings indirekte und eher für Sensornetzwerke geeignete Möglichkeit, im Rahmen der Datenerfassung Energie zu sparen, ist die Vorhersage von Werten, um so Kommunikationskosten zu vermeiden [135]. Dabei verwenden sowohl die Knoten als auch die Senke Modelle. Die Knoten erfassen trotz des Vorhersagemodells alle Werte und informieren bei zu großen Abweichungen die Senke, sodass beide Modelle angepasst werden wie beispielsweise in [136, 137].



Die Verwendung solcher Energiesparmaßnahmen macht die Vorhersage des Verbrauchs schwieriger, denn der tatsächliche Verbrauch hängt von externen Gegebenheiten ab, die sich nur grob abschätzen lassen. Ein Energiemanagement kann jedoch den Spielraum nutzen, den insbesondere die Verwendung unterschiedlicher Auflösungen des Sensors hinsichtlich des Energiebedarfs bietet, um den Systemverbrauch anzupassen und zu steuern. Der Umfang der Anpassungen ist allerdings von der Messgröße und dem verwendeten Sensor abhängig.

## 2.3 Messung versus Datenblatt

Bei der Entwicklung von tief eingebetteten Systemen ist eine wesentliche Frage, wie lange das System mit den gegebenen Anforderungen an die Servicequalität funktionieren kann. Neben dem Wissen um das Verhalten des Systems sind dafür zuverlässige Informationen über den Verbrauch notwendig. Die erste Quelle für diese Informationen ist das Datenblatt des Herstellers. Es enthält unterschiedlich detaillierte Werte über den Verbrauch in Bezug auf verschiedenste Gerätezustände, Versorgungsspannungen und Umweltbedingungen.

Wie bereits dargelegt, besteht ein tief eingebettetes System aus verschiedenen Modulen, die durch verschiedene analoge Bauteile ergänzt werden. Neben Varianzen dieser analogen und digitalen Komponenten können Herstellungsfehler auftreten. Diese Fehler müssen das System nicht unbrauchbar machen, können aber dessen energetischen Fußabdruck verändern.

Oftmals verwenden Ansätze, die sich mit dem Verbrauch von tief eingebetteten Systemen beschäftigen, Modelle, die mit Messungen von Knoten angereichert wurden. Leider verwenden die meisten dieser Ansätze nur einen Knoten, um ihr Modell aufzubauen, oder geben keine Information über die Anzahl der genutzten Knoten. In [138] wurde der Verbrauch von drei Mica2 [139] Sensorknoten gemessen. Festgestellt wurde eine Variation von rund 5% zwischen ihnen, ohne jedoch näher darauf einzugehen, woher diese stammen.

Der Einfluss von Leckströmen auf den Energieverbrauch im Schlafmodus, welche durch Variationen von Gate-Längen im CMOS-Herstellungsprozess verursacht werden, wurde in [140] mathematisch untersucht. Dabei wurde ein signifikanter Einfluss auf die Lebenszeit von Sensornetzen nachgewiesen.

In [141] wird eine bis zu 20-fache Abweichung bei dem für die Schlafmodi wichtigen Leckstrom in 230nm Technologie gezeigt. Hervorgerufen wird diese Abweichung durch Variationen in der Herstellung. In [142] wird der Einfluss von Gerätevariationen hinsichtlich der Schlafmodi analysiert. Für den untersuchten Atmel SAM3U [143], einen ARM Cortex M3 basierten  $\mu$ C, wurde eine Abweichung um den Faktor 5

zwischen den verschiedenen Knoten für den Verbrauch im Schlafmodus festgestellt. Für die aktiven Modi betrug die Abweichung rund 10%.

Da quantitative Analysen des Energieverbrauchs mit mehr Knoten bisher nicht bekannt sind, beschäftigt sich dieses Unterkapitel mit der Verlässlichkeit von Herstellerangaben und damit, inwieweit Abweichungen zwischen den Knoten existieren.

### 2.3.1 Messaufbau

In einer im Rahmen der Arbeit durchgeführten Messreihe wurde der Verbrauch von Texas Instruments eZ430-Chronos Entwicklungssystemen [144] ermittelt. Diese sind mit einem MSP CC430F6137 [54] Mikrocontroller ausgestattet. Der Controller beinhaltet ein 868MHz Funkmodul. Das System enthält zusätzlich einen Sensor für barometrischen Druck und einen Beschleunigungsmesser. Alle Komponenten sind direkt an die Batterie angeschlossen. Wie Abbildung 2.15a zeigt, verfügt das System über den Formfaktor einer Sportuhr und benötigt standardmäßig eine CR2032 Knopfzelle zur Energieversorgung. Zwar ist das Entwicklungssystem nicht explizit als Sensorknoten vorgesehen, allerdings scheint es durch seine Leistungsparameter, die Größe und den für dedizierte Sensorknoten vergleichsweise geringen Preis gut geeignet.

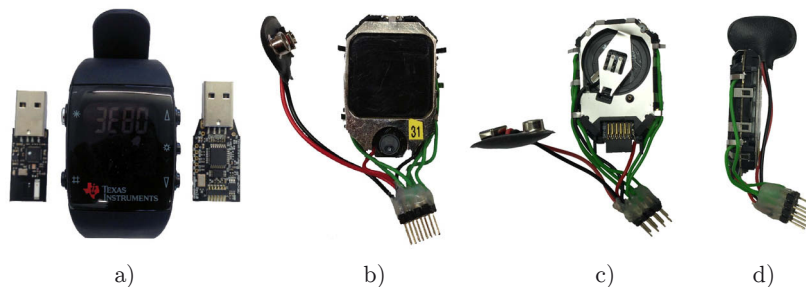


Abbildung 2.15: Untersuchter Sensorknoten a) Produktabbildung mit Gehäuse und mitgeliefertem Accesspoint (links) und Programmieradapter (rechts), b) modifizierter Knoten Front-, c) Rück- und d) Seitenansicht

Wie Abbildung 2.15 b) bis d) zeigen, sind die Knoten mittels Ersetzen der Taster durch Pins modifiziert worden. Dies geschah mit dem Ziel, Zugang zu den Ports zu erlangen, um weitere Peripherie anschließen zu können und eine flexiblere Energieversorgung zu ermöglichen.

Tabelle 2.7 zeigt die untersuchten Betriebsmodi. Die Betriebsmodi umfassen Schlaf- und Wachzustände des  $\mu\text{C}$ , aktive Zustände des Funkmoduls mit verschiedenen Parametern sowie die für die Erfassung analoger Werte notwendige Referenzspannungserzeugung und den Analog-Digital-Wandler.

90 Knoten wurden bei 3,0V und 2,2V mit 24 und 19 Modi gemessen. Dieser Unterschied resultiert aus den durch den  $\mu\text{C}$  bereitgestellten Energiestufen (eng. power level - PL), welche die mögliche Frequenz bestimmen. PL2 ist dabei der zu

Tabelle 2.7: Gemessene Betriebsmodi des MSP CC430F6137

Name	Beschreibung	
lpm3PL0 lpm3PL1 lpm3PL2 lpm3PL3	$\mu$ C Schlafmodus	
active@1MhzPL0 active@1MhzPL1 active@1MhzPL2 active@1MhzPL3	1MHz	$\mu$ C Aktivmodus
active@8MhzPL0 active@8MhzPL1 active@8MhzPL2 active@8MhzPL3	8MHz	
active@16MhzPL2 active@16MhzPL3	16MHz	
active@20MhzPL3	20Mhz	
RF idle	idle	CC1101 Funkmodul
RF intermediate	Zustandsübergang	
RF rx sensitive RF rx current	Empfang max. Sensitivität Empfang min. Verbrauch	
RF tx -6dBm RF tx 0dBm RF tx +10dBm	Sendemodus	
ref adc	Referenzgenerierung Analog-Digital-Wandler	

bevorzugende Modus, denn er ermöglicht alle Peripheriemodi und Frequenzen, ausgenommen einen Betrieb des  $\mu$ C mit 20MHz. Die Modi des Funkmoduls, des *adc* und *ref* beinhalten den Verbrauch für den  $\mu$ C Schlafmodus in PL2, welcher laut Datenblatt eine Stromaufnahme von 2,2 $\mu$ A bei 3V hat. Um die ermittelten Werte mit dem Datenblatt vergleichen zu können, wurden die Modi genau wie für die Referenzwerte beschrieben konfiguriert. Abweichungen sind allerdings zu erwarten, da die verwendeten Knoten über mehr Peripherie und Bauteile verfügen als die Testexemplare beim Hersteller.

Der in Abbildung 2.16 dargestellte Messaufbau bestand aus einer PowerScale Einheit [145], die zusammen mit einem Messfühler in der Lage ist, in einem dynamischen Bereich von 200nA bis 500mA bei 100k Samples/sek Strom und Spannung zu messen. Dies ist durch die dynamische und in kürzester Zeit erfolgende Umschaltung zwischen sieben verschiedenen Messwiderständen im Messfühler möglich.

Zusätzlich ist das Messgerät in der Lage, UART-Ereignisse von dem untersuchten Gerät zu empfangen. Für die Energieversorgung wurde ein lineares Labornetzteil [146] genutzt, welches zusätzlich mit einem Tiefpassfilter ausgestattet wurde, um das Rauschen weiter zu verringern.

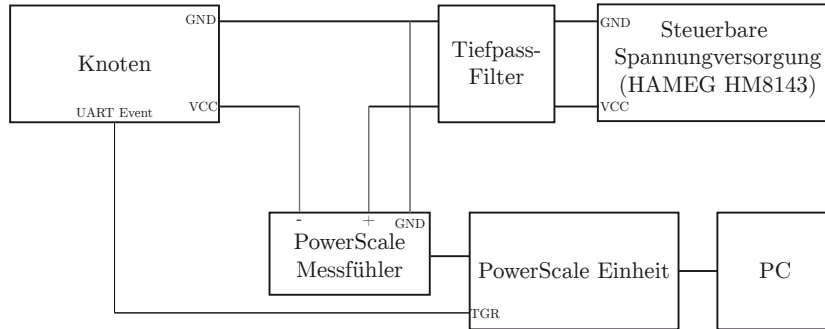


Abbildung 2.16: Schematischer Messaufbau mit Spannungsversorgung, Messgerät und untersuchtem Knoten

Da die Temperatur einen Einfluss auf den Verbrauch haben kann, wurden die Messungen in einem Raum mit Klimaanlage durchgeführt, um diesen Einfluss so weit wie möglich auszuschließen. Die Temperatur lag bei 25°C und fluktuierte um weniger als 1°C.

Um die Messgenauigkeit zu ermitteln, wurde die Standardabweichung  $\sigma$  für jeden Knoten in jedem Modus berechnet. Wie erwartet, steigt  $\sigma$  mit zunehmender Leistungsaufnahme, was im Aufbau des Messgerätes mit verschiedenen Messwiderständen begründet ist. Im Schlafmodus bei einer Stromaufnahme von unter 8µA liegt  $\sigma$  bei 0,09µA und steigt bis auf 87,13µA im Modus *RF tx +10dBm* bei einem Verbrauch von rund 34mA. Im Verhältnis zu den Messwerten betrachtet, sinkt mit steigender Stromaufnahme die relative Messabweichung. Für die Modi mit einem Verbrauch größer als 3mA liegt sie unter 0,25%, während sie für die anderen Modi unter 0,6% liegt. Ausgenommen davon sind der Schlafmodus mit einer Abweichung von durchschnittlich 3% sowie Ausreißer, die sich entsprechend im Verbrauch zeigen.

### 2.3.2 Messergebnisse

Abbildung 2.17 zeigt die gemessene durchschnittliche Stromaufnahme aller Knoten in mA für alle Modi gemeinsam in einer Skala. Der Verbrauch reicht über mehrere Größenordnungen von einigen µA bis zu 35mA. Der Sendemodus des Radios mit +10dBm benötigt 10.000 mal mehr Energie als der µC Schlafmodus. Um lange Standzeiten zu erreichen, verbringen tief eingebettete Systeme und allen voran Sensorknoten, jedoch viel mehr Zeit schlafend als mit Sendevorgängen.

Wie Abbildung 2.18 zusammen mit Tabelle B.1 im Anhang zeigt, steigt der absolute Abstand zwischen den Knoten mit niedrigstem und höchstem Verbrauch bei

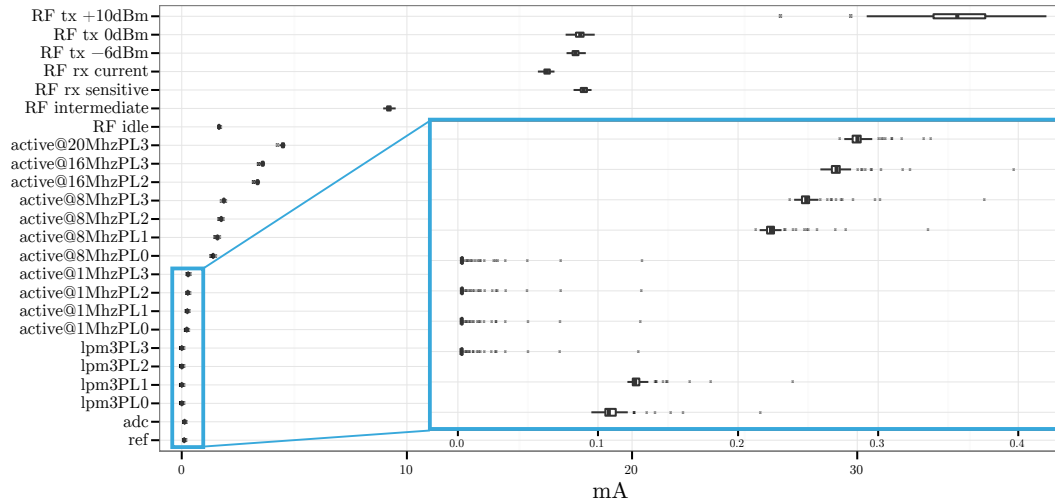


Abbildung 2.17: Verteilung der gemessenen Stromaufnahme als Boxplot aller Knoten für alle betrachteten Modi bei 3,0V

Modi mit erhöhter Stromaufnahme, wohingegen die relative Abstand sinkt und die Zahl der Ausreißer kleiner wird. Einige Knoten haben einen außergewöhnlich hohen Verbrauch in Modi mit geringer Stromaufnahme. Beispielsweise gibt es einen Knoten, der 46 mal so viel Energie im Schlafmodus verbraucht wie der Median aller Knoten. Diese extremen Ausreißer zeigen, dass der Durchschnitt nicht ausreicht, um die Leistung der Knoten zu bewerten.

Die Stromaufnahme variiert besonders stark in den Schlafmodi *lpm3PL0* bis *lpm3PL3*. Beispielsweise liegt der Median bei 3,0V im LPM3 mit PL2 bei 2,835 $\mu$ A, wobei sich der Verbrauch zwischen 2,53 $\mu$  und extremen 131,4 $\mu$ A bewegt. Auch bei 2,2V ist die Stromaufnahme noch stark gestreut und liegt zwischen 2,1 $\mu$  und 32,9 $\mu$ A. Dies zeigt, dass einige Knoten während des Schlafes einen abnormal hohen Verbrauch haben. Für einige dieser Knoten fluktuiert die Stromaufnahme zusätzlich, erkennbar durch ein deutlich größeres  $\sigma$ , während diese bei den anderen Knoten im Rahmen der Messgenauigkeit stabil ist. Dies sind Indikatoren dafür, dass die Knoten scheinbar unter nicht funktionalem Aspekt defekt sind, da sich der Verbrauch deutlich außerhalb der Spezifikation befindet, obwohl sie sich ansonsten normal verhalten. Allerdings würde ein Einsatz dieser Knoten in einem produktiven Umfeld zu einem unerwartet frühen Ausfall durch entleerte Batterien führen.

Der Verbrauch und Umfang der Abweichung zwischen den Knoten mit der höchsten und der niedrigsten Stromaufnahme hängt deutlich von der Versorgungsspannung ab. Bei 2,2V sind der Unterschied weit geringer und der Verbrauch generell niedriger. Dies gilt im Wesentlichen für alle gemessenen Modi, wobei für das Funkmodul nur eine geringe bis gleichbleibende Differenz besteht.

Für die aktiven Modi des  $\mu$ C liegt der geringste Unterschied zwischen den Knoten mit der höchsten und der niedrigsten Stromaufnahme bei 7% bei 20MHz Takt. Mit

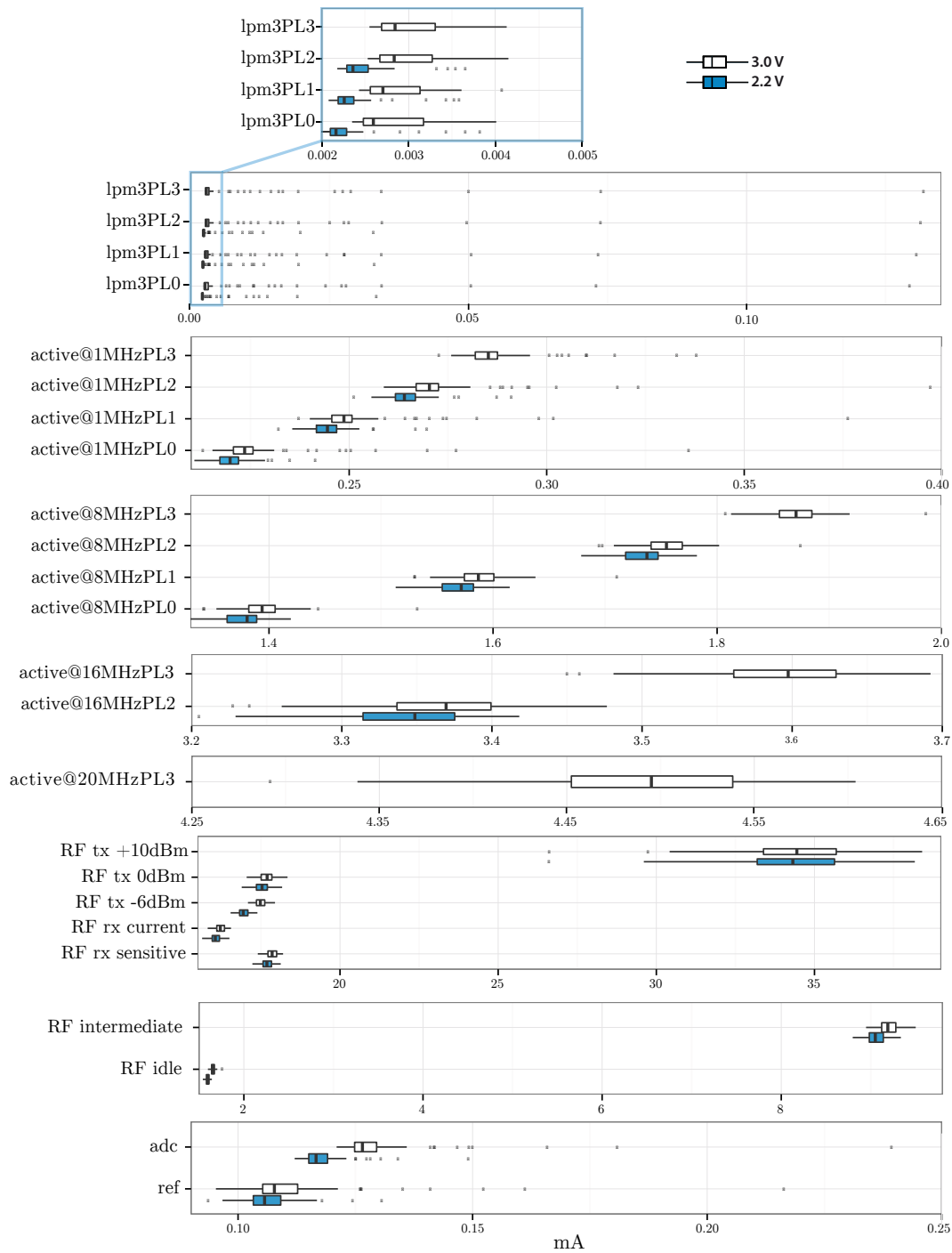


Abbildung 2.18: Detaillierte Verteilung der gemessenen Stromaufnahme als Boxplot aller Knoten, aufgeteilt nach Betriebsmodi für 2,2V und 3,0V

sinkendem Takt und geringerer Stromaufnahme steigt diese Streuung bis auf 58% bei 1MHz und 3,0V.

Der Einfluss der durch den  $\mu\text{C}$  bereitgestellten Energiestufen ist in Abbildung 2.18 erkennbar. Grundsätzlich nimmt die Stromaufnahme mit steigender Energiestufe zu. Während der Einfluss im Schlafmodus noch gering ist, verbrauchen die aktiven Modi deutlich mehr Energie. Die Stromaufnahme von PL0 und PL3 unterscheidet sich um rund 30%.

Die durchschnittliche Stromaufnahme des ADC differiert zwischen dem besten und dem schlechtesten Knoten um 98% bei 3,0V und 33% bei 2,2V. Für die interne Referenzspannungserzeugung ist der Unterschied mit 124% und 40%, noch größer. Diese hohe Abweichung liegt im Verbrauch des Schlafmodus begründet, welcher, wie bereits dargelegt, für manche Knoten abnormal hoch ist.

Je nach Betriebsmodus unterscheiden sich die besten und die schlechtesten Knoten in Bezug auf den Verbrauch des Funkmoduls um 5-10%. Eine Ausnahme ist der Sendemodus mit 10dB, in dem einige Knoten mehr als der Median und andere deutlich weniger als der Median verbrauchen. Somit liegt der Unterschied bei  $\sim 44\%$  zwischen bestem und schlechtestem Knoten.

Bei genauerer Untersuchung wurde die Antenne als Ursache des Problems identifiziert, denn bei einigen Knoten war die Antenne lose. Diese Knoten hatten eine deutlich geringere Stromaufnahme als angenommen. Als Antenne fungiert das in Abbildung 2.15 b) zu erkennende Blech am Displayrand. Ein ähnliches Verhalten ließ sich an Knoten feststellen, deren ursprüngliche Antenne durch eine einfache drahtbasierte Antenne ausgetauscht wurde, die zusätzlich die Reichweite merklich erhöhte. Allerdings sind diese modifizierten Knoten nicht Teil der Messreihe.

Die teilweise deutliche Abweichung zwischen den Knoten wirft die Frage auf, ob es „gute“ und „schlechte“ Knoten gibt, also Knoten, die in allen Modi einen geringeren beziehungsweise höheren Verbrauch haben. Die Abbildungen 2.19 und 2.20 zeigen die durchschnittliche Stromaufnahme eines jeden Knotens (horizontal) in jedem gemessenen Betriebsmodus (vertikal) im Verhältnis zum Median über alle Knoten. Die Farbintensität verdeutlicht den Abstand zum Median, wobei ein „-“ (grün) anzeigt, dass dieser Knoten besser ist als der Median über alle Knoten, und ein „+“ (rot), dass er schlechter ist. Die Abbildungen zeigen, dass es Knoten gibt, die ausnahmslos gut oder schlecht sind. Allerdings gilt dies nur für einen kleinen Teil. Für die Mehrzahl der Knoten variieren gute und schlechte Modi. Es ist keine Vorhersage möglich, ob ein in einem Modus guter Knoten in einem anderen Modus zwingend gut oder schlecht sein wird.

Die beobachteten Abweichungen und deren Abhängigkeit von der Versorgungsspannung und Stromaufnahme können verschiedene Ursachen haben. Eine Abweichung von den Datenblattwerten ist insbesondere im Schlafmodus durch die zusätz-

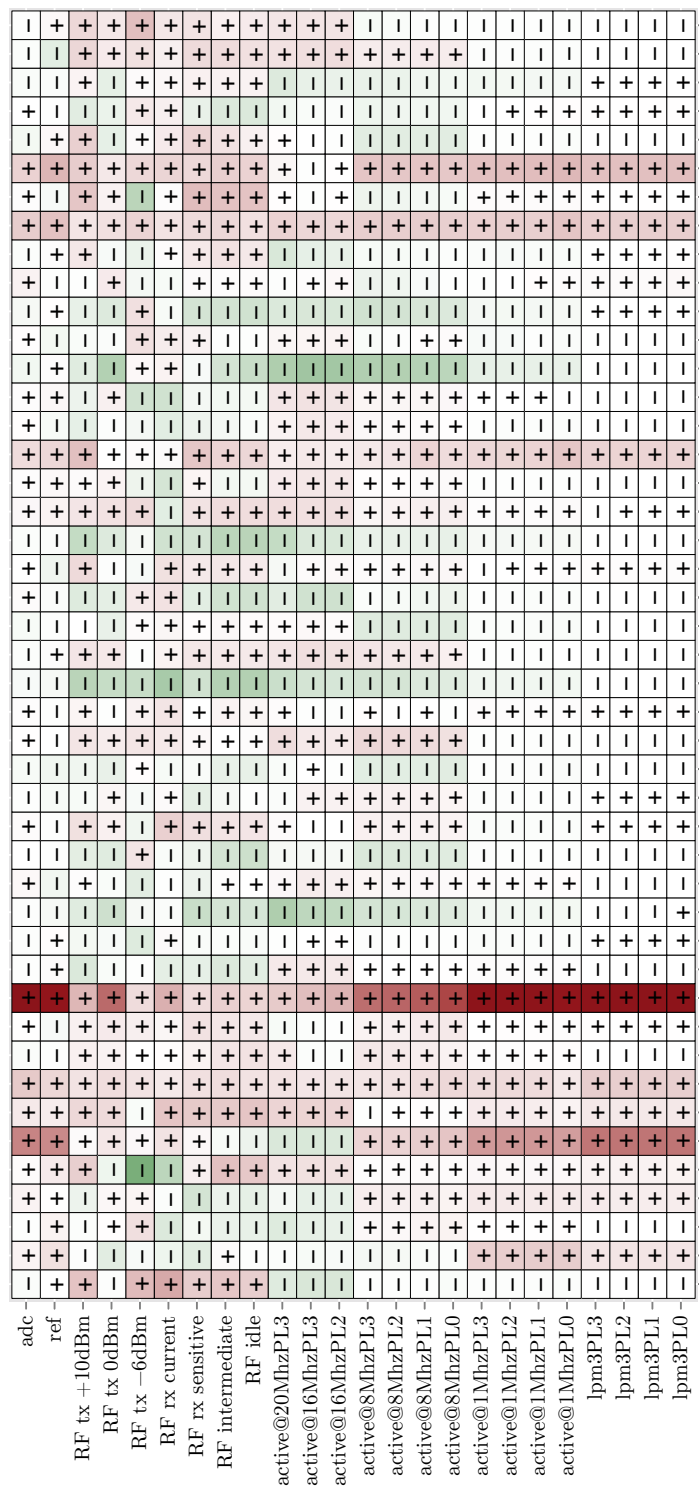


Abbildung 2.19: Durchschnittliche Stromaufnahme jedes Knotens (horizontal) in einem Betriebsmodus (vertikal): Die Farbintensität bestimmt den Abstand zum Median über alle Knoten. Ein ‘-‘ zeigt an, dass dieser Modus weniger verbraucht als der Median, und ein ‘+’ dass dieser mehr verbraucht. (Teil 1)



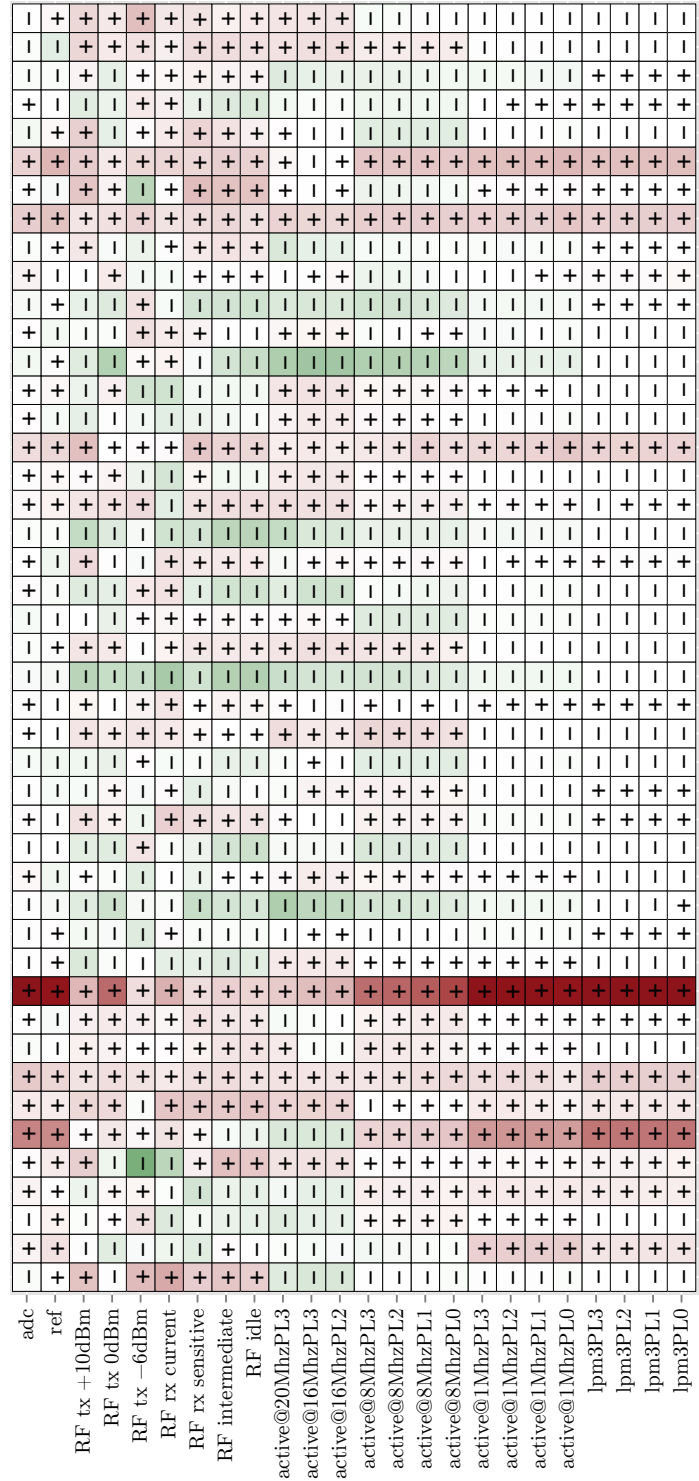


Abbildung 2.20: Durchschnittliche Stromaufnahme jedes Knotens (horizontal) in einem Betriebsmodus (vertikal): Die Farbintensität bestimmt den Abstand zum Median über alle Knoten. Ein ‘-‘ zeigt an, dass dieser Modus weniger verbraucht als der Median, und ein ‘+‘ dass dieser mehr verbraucht. (Teil 2)

lichen Bausteine zu erwarten. Allerdings verbrauchen rund 20% der Knoten deutlich mehr Energie und haben ein deutlich höheres  $\sigma$ . Der generelle Aufbau des Knotens mit der Blechantenne kann hierfür eine Ursache sein. Ein weiterer möglicher Einflussfaktor kann in unbemerkten Fehlern bei der Modifikation der Knoten liegen. Des Weiteren kann der taktgebende Quarz Fertigungsschwankungen unterliegen und nicht die erwarteten 32kHz liefern. Werden die 19 vom sehr hohen Verbrauch betroffenen Knoten aus den Daten entfernt und als defekt angenommen, verbessert sich die Streuung der Knoten deutlich. Im Schlafmodus unterscheiden sich bei 3,0V der schlechteste ( $4,15\mu\text{A}$ ) und beste Knoten ( $2,1\mu\text{A}$ ) nur noch um 64%, was zwar deutlich geringer, aber immer noch signifikant ist. Diese Verbesserung trifft im selben Maße auch auf alle anderen Modi zu, da die „defekten“ Knoten in den meisten Modi die Ausreißer darstellen.

### Vergleich zwischen Messungen und Datenblatt

Abbildung 2.21 zeigt die gemessene Stromaufnahme aller Knoten im Vergleich zu den Datenblattwerten für 3,0V. Daten für 2,2V sind nicht verfügbar. Die gemessene Stromaufnahme wird mit den typischen Werten und, sofern angegeben, mit den maximalen Werten des Datenblattes verglichen.

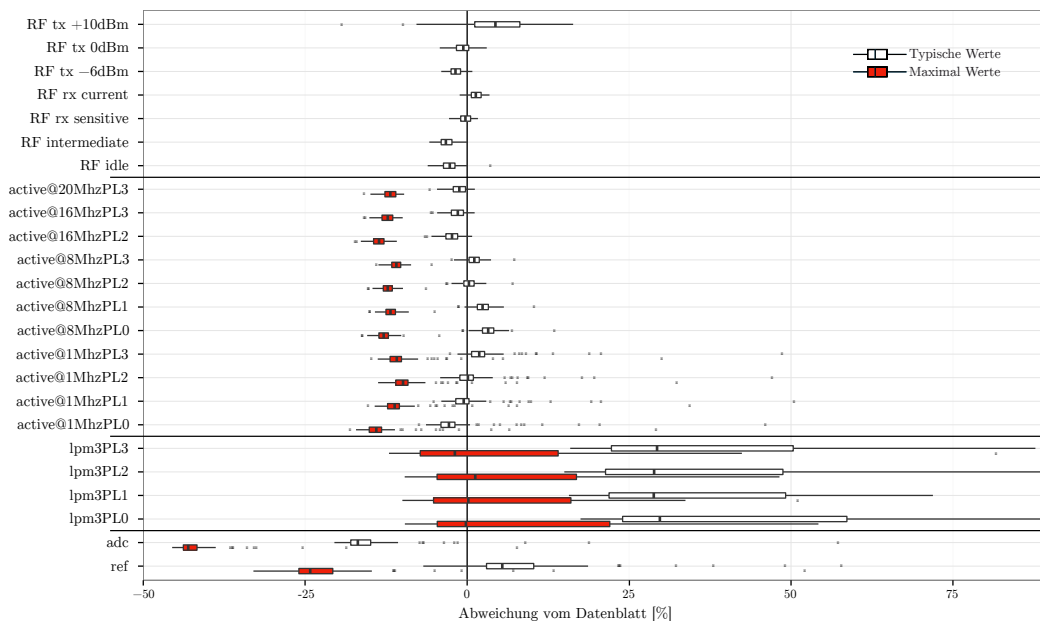


Abbildung 2.21: Relative Abweichung der Messergebnisse von den durchschnittlichen und maximalen Datenblattwerten bei 3,0V

Da bei den Knoten mehr externe Komponenten vorhanden sind als bei der Ermittlung der Datenblattwerte, können deren Bedingungen nicht vollständig nachgebildet werden. Ein höherer Verbrauch ist deshalb natürlich und zu erwarten. Allerdings ist

die Abweichung vom Datenblatt in den Schlafmodi sehr hoch und weit gestreut. 20% der Knoten konnten in der Abbildung nicht dargestellt werden, da sie ein Vielfaches des Erwartungswertes verbrauchten. Im Vergleich mit der im Datenblatt angegebenen maximalen Stromaufnahme benötigen nur noch rund die Hälfte der Knoten mehr Strom im Schlafmodus. Allerdings führen die maximalen Werte zu einer deutlich zu pessimistischen Abschätzung des Verbrauchs in allen anderen Modi.

Für diese ist die Abweichung zwischen Messung und typischen Datenblattwerten deutlich niedriger und liegt zum Teil unter den Erwartungswerten. Mit steigender Stromaufnahme sinken die Streuung und die Zahl der Ausreißer.

Abhängig vom Knoten können sich diese Abweichungen gegenseitig abschwächen, wie beispielsweise bei der Referenzspannungsgenerierung und dem ADC. Zwar benötigt *ref* mehr Energie als erwartet, wird aber meist zusammen mit dem ADC genutzt, sodass der höhere Verbrauch relativiert wird.

### **Einfluss der Abweichungen**

Der Einfluss der gezeigten Abweichungen hängt davon ab, wie lange ein Knoten in einem Modus verweilt. Ein Knoten, der die meiste Zeit schläft und nur selten Nachrichten übermittelt, wird stärker unter einer erhöhten Stromaufnahme des Schlafmodus leiden als ein Knoten, der häufig lauscht und viele Nachrichten sendet.

Tabelle 2.8: Stromaufnahme vom Beispielknoten

	Durchschnittliche Stromaufnahme in mA			Aktive Zeit ms
	Datenblatt	Knoten A	Knoten B	
LPM3 sleep	0,0022	0,0027	0,0042	-
CPU@16Mhz	3,451	3,311	3,378	3
arc + ref	0,2522	0,225	0,224	0,2
Radio RX	16,0022	15,817	16,338	40
Radio TX@10dBm	33,0022	31,159	34,051	5

Um den möglichen Einfluss zu demonstrieren, wurden die Verbrauchswerte zweier zufällig gewählter Knoten (keine Ausreißer) mit denen des Datenblattes in einer theoretischen zyklischen Messanwendung verglichen. Während der aktive Energieverbrauch gleich bleibt, ändert sich der Anteil des Schlafmodus mit der Periode. Tabelle 2.8 zeigt die gemessene Leistungsaufnahme der beiden Knoten und des Datenblatts zusammen mit dem zeitlichen Verhalten, das zur Konstruktion des Verbrauchsmodells der Anwendung genutzt wurde.

Der Unterschied zwischen der Annahme, basierend auf Datenblattwerten und tatsächlichen Verbrauchswerten, ist in Abbildung 2.22 mit einer variablen Periode dargestellt. Während Knoten A bis zu einer Periode von einer Minute minimal we-

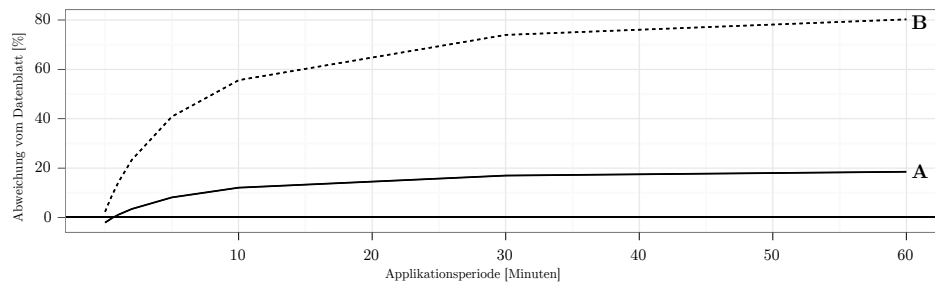


Abbildung 2.22: Prozentuale Abweichung des angenommenen vom tatsächlichen Verbrauch bei unterschiedlichen Applikationsperioden

niger als angenommen verbraucht, benötigen beide Knoten bei steigender Periode mehr Energie, als nach den Werten des Datenblattes erwartet. Dies ist im sinkenden Einfluss der zum Teil weniger als erwartet verbrauchenden aktiven Modi begründet, da der Schlafmodus mit steigender Periode immer dominanter wird. Obwohl die absolute Abweichung bei Knoten B im Schlafmodus nur bei rund  $2\mu\text{A}$  liegt, wirkt sich dieser Wert im Betrieb mit einer Abweichung von bis zu 80% aus. Das zeigt, dass auch eine sehr kleine Abweichung zu großen Veränderungen führen kann.

Als Konsequenz werden die Vorhersagen, welche auf dem Datenblatt basieren, umso ungenauer, je mehr Energie der Entwickler zu sparen versucht.

### 2.3.3 Mögliche Konsequenzen

Folgt man der Vision der Sensornetze, die aus tausenden billigen Knoten bestehen sollen, erscheint es nicht als angemessen, den Netzwerkentwickler mit der Messung eines jeden einzelnen Knotens zu belasten. Um das zu vermeiden, könnte der Knotenhersteller Garantien über einen maximalen Verbrauch geben, was sich durch eine sorgsame Auswahl der Komponenten und Testen jedes Knotens realisieren ließe. Allerdings würde das die Herstellungskosten deutlich erhöhen. Das steht im Widerspruch zum Ziel, Hardware möglichst günstig zu produzieren. Eine alternative Möglichkeit ist die Nutzung pessimistischer Abschätzungen und Energiemodelle, die eine möglichst große Anzahl an Knoten abdecken. Zusätzlich muss adaptives Energiemanagement betrieben werden, um größere Abweichungen während des Betriebes auszugleichen.

Neue Technologien können den Energieverbrauch im Schlafmodus zukünftig reduzieren. So ist es denkbar, Mikrocontroller mit NVRAM Technologie [147, 148] auszustatten, was in Kombination mit nA Wake-up Timern z. B. [149] das Abschalten des gesamten Controllers ermöglicht.

### 2.4 Zusammenfassung

In diesem Kapitel wurden die Einflussfaktoren auf die Energieversorgung und den Energieverbrauch in drahtlosen tief eingebetteten Systemen betrachtet. Dabei wurde deutlich, dass nicht nur die Software mit ihrem unterschiedlichen Verhalten bei der Steuerung von Kommunikation und Datenerfassung einen Einfluss auf den Verbrauch hat, sondern auch die Hardware selbst. Die aus einer Batterie entnehmbare Energiemenge hängt sowohl von der Chemie der Zellen ab, als auch von der Umgebungstemperatur und dem Entladeverhalten. Durch Fertigungsschwankungen auf der einen Seite und eine Verwendung von Spannungswandlern auf der anderen Seite kann der tatsächliche Energieverbrauch des eingebetteten Systems vom erwarteten Verbrauch abweichen. In einer Messreihe wurden die Schwankungen für einen Typ Sensorknoten quantifiziert und gezeigt, inwieweit Annahmen auf Basis des Datenblatts mit den tatsächlichen Werten übereinstimmen.

Diese Faktoren müssen durch ein Energiemanagement adressiert werden. Hierzu müssen der Verbrauch erfasst und Abweichungen durch Überwachung der Batteriekapazität festgestellt werden, um im Bedarfsfall durch geeignete Maßnahmen steuernd einzugreifen.



## Kapitel 3

# Ladezustandsüberwachung zum Erreichen von Lebenszeitzielen

Die Energiespeicher stellen einen limitierenden Faktor für die Laufzeit und damit für die Leistung drahtloser tief eingebetteter Systeme dar. Im Allgemeinen werden zur Versorgung Batterien und Akkumulatoren eingesetzt, da diese eine vergleichsweise hohe Energiemenge speichern können und so eine Versorgung über einen längeren Zeitraum ermöglichen. Wie in Kapitel 2.1.1 dargelegt, unterliegen Batterien und Akkumulatoren allerdings nichtlinearen Effekten, welche die Kapazität einer Zelle von deren Verwendung abhängig machen. Das kann dazu führen, dass die entnehmbare Energiemenge deutlich von der Herstellerspezifikation abweicht. Hinzu kommt eine Temperaturabhängigkeit. Letzteres kann gerade bei tief eingebetteten Systemen problematisch sein, die Umweltbeobachtungen durchführen, denn sie sind den Umweltbedingungen voll ausgesetzt.

Sinkt die Spannung der Batterie oder eines Akkumulators temperaturbedingt unter einen Wert, der vom System verarbeitet werden kann, kann das System ausfallen. Eine Überwachung des Ladezustands ist somit notwendig, um gegebenenfalls auf unerwartetes Verhalten in Bezug auf die Kapazität oder Spannung reagieren zu können.

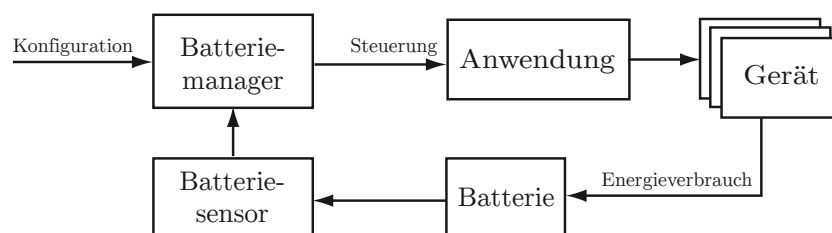


Abbildung 3.1: Abstrakte Regelungsschleife zur Steuerung der Anwendung auf Grundlage des Batteriezustandes

Abbildung 3.1 stellt den grundsätzlichen Ansatz zur Steuerung der Anwendung auf Grundlage des Batteriezustandes dar. Die Anwendung verbraucht durch Nutzung der Geräte des eingebetteten Systems Energie der Batterie. Dies wird vom System observiert und führt gegebenenfalls zur Anpassung der Anwendung.

In diesem Kapitel wird ein neuer erweiterter Ansatz für die Erfassung des Ladezustandes vorgestellt, denn diese Information ist vital für ein Energiemanagement, um den Verbrauch des Systems steuern zu können. Wie in Kapitel 1 motiviert, ist es für Anwendungen mitunter nicht zielführend, eine möglichst lange Standzeit zu erreichen. Für derartige Anwendungen ist das Erreichen einer vorgegebenen Laufzeit notwendig. Eine solche Vorgabe ermöglicht es, die verbleibende Energie so einzuteilen, dass diese bis zum Laufzeitziel ausreicht und dadurch eine möglichst homogene Anwendungsqualität erreicht wird.

### 3.1 Problembeschreibung

Jedwede Aktivität eines drahtlosen tief eingebetteten Systems verbraucht Energie. Die daraus entstehende Auswirkung auf die Energiequelle ist jedoch nicht notwendigerweise sofort beobachtbar. Es bietet sich daher an, die in der Quelle verbleibende Energie nicht kontinuierlich, sondern in diskreten Intervallen zu ermitteln. Die gewünschte Lebenszeit eines Systems lässt sich hierzu in  $T$  diskrete Zeitschritte einteilen. Dabei ist Zeitschritt  $T - 1$  das Ziel, dessen Ende erreicht werden muss.

Um ein festgelegtes Lebenszeitziel zu erreichen, muss die zur Verfügung stehende Energie verwaltet werden. Dazu benötigt der Energiemanager die Information über die verbleibende Energie  $E(t)$  in einem spezifischen Zeitschritt  $t$ . Die in jedem folgenden Zeitschritt verfügbare Energie lässt sich, wie in Abbildung 3.2 dargestellt, leicht aus verbleibender Zeit und verbleibender Energie berechnen:  $E(t)/(T - t)$ .

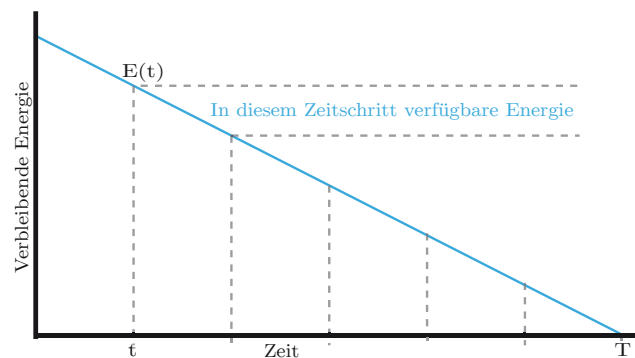


Abbildung 3.2: Ermittlung der in einem Zeitschritt zur Verfügung stehenden Energie



Allerdings ist die Information über den Ladezustand und damit der verbleibenden Energie bei galvanischen Zellen ein nichttriviales Problem. Andere Energiespeicher wie Superkondensatoren verhalten sich linearer und eine Aussage über den Ladezustand ist einfacher.

### 3.2 Stand der Technik

Techniken zur Ermittlung des Ladezustandes von Batterien und Akkumulatoren lassen sich grob in drei Bereiche einteilen. Hardware-basierte Ansätze (Kapitel 3.2.1) nutzen dedizierte Batteriemanagement-Co-Prozessoren. Modell-basierte Ansätze (Kapitel 3.2.2) verwenden unter anderem physikalische oder stochastische Batteriemodelle, um das Verhalten der Batterie nachzubilden. Messbasierte Ansätze (Kapitel 3.2.3) nutzen beispielsweise aufgezeichnetes Entladeverhalten.

Im Folgenden werden Vertreter der jeweiligen Bereiche vorgestellt und hinsichtlich ihrer Kosten und ihres Nutzens bewertet.

#### 3.2.1 Hardware-basierte Ansätze

Hardware-basierte Ansätze verursachen immer eine Erhöhung der Fertigungskosten und Komplexität des Systems. Zusätzlich verbrauchen sie Energie, was bei der Auswahl bedacht werden muss. Ihr Vorteil ist allerdings, dass nahezu kein Verarbeitungsaufwand durch Software anfällt, da alle Berechnungen direkt durch Hardware erfolgen.

#### Smart Battery und Ladezustandsmesser

Diese Klasse von Co-Prozessoren wird oftmals direkt im Batteriepack eingesetzt und verbleibt dort. Das ist insbesondere dann sinnvoll, wenn der Co-Prozessor in der Lage ist, das Alter und den Gesundheitszustand des Akkumulators zu erfassen. Das Smart Battery System (SBS)<sup>1</sup> bezeichnet dabei keine einzelne Hardwarelösung, sondern vielmehr einen in den Advanced Configuration and Power Interface (ACPI)<sup>2</sup> integrierten Standard, um einem Betriebssystem Informationen über den Ladezustand und Verbrauch geben zu können. Für diesen Zweck werden Batteriemanagement-Co-Prozessoren eingesetzt, welche über einen spezialisierten Bus<sup>3</sup> angesprochen werden. Viele Energiemanagement-Ansätze basieren auf der Nutzung der durch ACPI bereitgestellten Informationen.

---

<sup>1</sup>siehe <http://smartbattery.org/>

<sup>2</sup>siehe <http://acpi.info/>

<sup>3</sup>siehe <http://smbus.org/>

Um den Ladezustand zu ermitteln, nutzen solche Co-Prozessoren den Spannungsabfall an einem Messwiderstand durch einen integrierten Analog-Digital-Wandler und erfassen so die entnommene und die geladene Energiemenge. Dieses Prinzip wird auch als sogenannte Coulomb-Zählung bezeichnet. Ein solches Verfahren hat allerdings Nachteile, was die Auflösung und die Bandbreite betrifft. Je nach verwendetem Widerstand liegt die kleinste zu erfassende Einheit im Bereich von einigen Milliampere. Das macht solche Co-Prozessoren für tief einbettete Systeme unattraktiv, die nur einen geringen Stromverbrauch besitzen, da dort ein Bereich von wenigen Mikroampere bis hundert Milliampere verwendet wird. Ein weiterer limitierender Faktor ist die Samplingfrequenz des AD-Wandlers, denn diese bestimmt, wie schnell Änderungen im Verbrauch erfasst und berücksichtigt werden können. Zusätzlich entstehen durch den Co-Prozessor Energiekosten im Bereich von hundert Mikroampere, die deutlich über den Kosten für den Schlafzustand eines drahtlosen tief eingebetteten Systems liegen. Der große Vorteil dieses Verfahrens ist allerdings, dass neben den Informationen über den Ladezustand die verbrauchte Energiemenge erfasst wird und sich dieser Wert für das Energiemanagement nutzen lässt. Bei der reinen Coulomb-Zählung müssen die Parameter der Batterie bekannt sein. Zusätzlich muss periodisch Aufwand für eine Rekalibrierung betrieben werden [150].

Es gibt verschiedene Co-Prozessoren für unterschiedliche Akkumulatortypen. Allerdings werden meist nur ein oder zwei Typen von einem System vollständig unterstützt. Die Berechnung des Ladezustands wird nicht von allen Geräten durchgeführt, insbesondere nicht von reinen Coulomb-Zählern, sondern teilweise dem Hostsystem überlassen, wie beispielsweise beim Smart Battery Monitor DS2438 [151]. Im Gegensatz dazu berechnet der Ladezustandsanzeiger BQ26500 [152] zusätzlich den Ladezustand für Li-Ion and Li-Pol Akkumulatoren.

#### **MAX17043**

Im Unterschied zu den anderen Co-Prozessoren erfasst der MAX17043 [153] nicht die Lade- und Entladeaktivität, sondern nutzt die Spannung, um den Ladezustand eines Li-Ion-Akkumulators zu bestimmen. Dabei werden spezielle Algorithmen verwendet, die sich durch einen Kompensationsparameter in eingeschränktem Maße anpassen lassen, wobei die notwendigen Informationen für dessen Berechnung nicht frei verfügbar sind. Die Spannung zu überwachen hat den Vorteil, dass Ungenauigkeiten durch die eingeschränkte Auflösung des Messverfahrens umgangen werden können. Allerdings ist damit auch keine Information über den Energieverbrauch verfügbar. Ein weiterer Vorteil ist, dass das System in einen Schlafzustand wechseln kann, wenn aktuell keine Daten benötigt werden, ohne dass wie bei den schon vorgestellten Systemen Informationen verloren gehen.

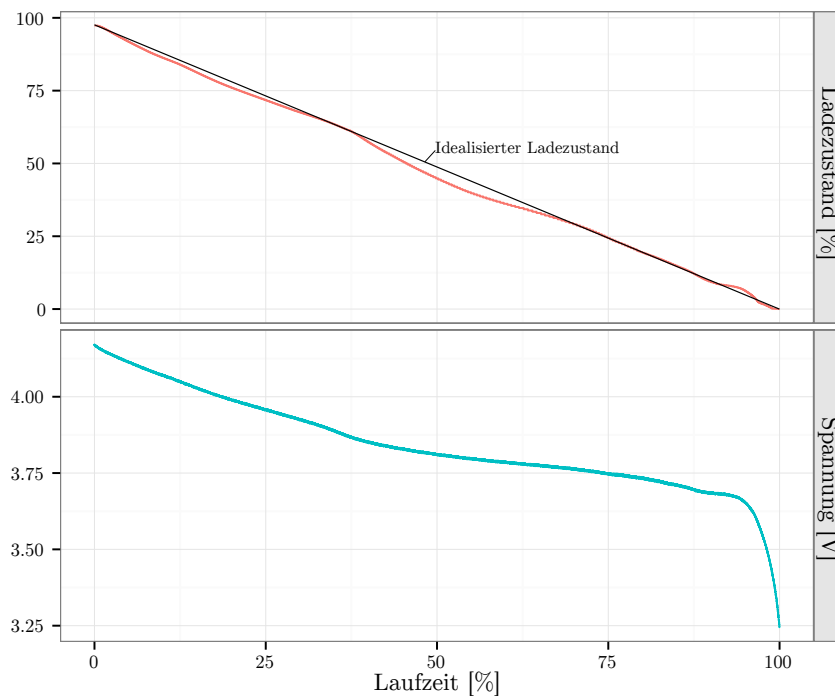


Abbildung 3.3: Experiment mit MAX17043 zur Ermittlung des Ladezustandes

Abbildung 3.3 zeigt den Ladezustand und die Spannung eines im Rahmen dieser Arbeit durchgeführten Experimentes mit einem MAX17043. Dabei wurde ein 110mAh Li-Ion-Akkumulator [154] innerhalb von 52 Stunden entladen. Der Spannungsverlauf des Akkumulators ist weitgehend linear, allerdings fällt die Spannung gegen Ende stark ab. Im Vergleich mit der eingezeichneten Ideallinie zeigt sich, dass der Ladezustand zwar nicht exakt ermittelt wurde, aber trotz Abweichungen hinreichend genau und ohne starke Schwankungen abgebildet wurde. Veränderungen in der Spannung schlugen sich direkt im Ladezustand nieder, wobei insbesondere der Spannungsabfall gegen Ende nur geringe Auswirkungen hatte.

### 3.2.2 Modell-basierte Ansätze

Es existiert eine Vielzahl an Ansätzen, die das Verhalten galvanischer Zellen nachbilden, um so den Ladezustand zu ermitteln. Dies geschieht auf unterschiedlichste Weise und mit verschiedener Komplexität. Allen gemein ist, dass sie einen hohen Aufwand erfordern, der die meisten tief eingebetteten Systeme überfordert. Dadurch sind diese Ansätze in der Regel ungeeignet.

#### Physikalische Ansätze

Diese Art der Modelle bildet das Verhalten von Batterien und Akkumulatoren am genauesten nach. Allerdings ist der zu betreibende Aufwand sehr hoch und die Modelle benötigen viele Parameter, die zum Beispiel die Struktur oder die chemische

Zusammensetzung der Zellen beschreiben. In [155] wird ein elektrochemisches Modell genutzt, um eine Lithium-Polymer-Zelle zu simulieren. Li-Ion-Akkumulatoren können, wie in [156] beschrieben, simuliert werden.

### **Äquivalenz-Schaltungen**

Ansätze, die auf Äquivalenz-Schaltungen basieren, modellieren das Verhalten von Batterien abstrakt mithilfe elektrischer Bauteile, wie Kapazitäten, Widerständen und Spannungsquellen. Daneben können je nach Anforderung auch Tabellen mit vorher ermittelten Werten verwendet werden. Der Aufwand dafür ist geringer als bei physikalischen Modellen. Die Modelle können sowohl kontinuierlich wie bei [157] als auch diskret wie bei [158] sein. Zur Verbesserung der Genauigkeit wird in [159] ein hybrides Modell mit Kalman-Filter verwendet.

### **Stochastische Ansätze**

In [160] und erweitert in [161] werden diskrete transiente stochastische Prozesse genutzt, um Entladung und Erholung einer Zelle bei pulsformiger Last abzubilden. Zwar ist der Berechnungsaufwand geringer als bei den vorhergehenden Modellen, allerdings müssen notwendige Parameter wie die Stärke der Erholung und die nominale Kapazität der Zelle durch physikalische Simulation ermittelt werden.

### **Neuronale Netze**

Künstliche neuronale Netze können wie in [162] und [163] genutzt werden, um nach einer Trainingsphase aus Spannung, Temperatur und Last den Ladezustand zu ermitteln. Sie können dadurch für unterschiedliche Zellentypen eingesetzt werden, was sie im Vergleich zu den anderen Modell-basierten Ansätzen flexibler macht. Die Komplexität und der Laufzeitaufwand hängen dabei stark von der Anzahl der für das Modell notwendigen Neuronen ab.

### **Fuzzylogik**

Fuzzylogik lässt sich, wie beispielsweise in [164] diskutiert, nutzen, um gesammelte Daten zu analysieren und so Modelle zur Vorhersage des Ladezustands zu generieren. In [165] wurde mithilfe von Fuzzylogik aus Impedanz und Spannungsdaten ein Modell für Li-Ion Akkumulatoren entwickelt, das in der Lage ist, die Anzahl verbleibender Pulse eines mobilen Defibrillators vorherzusagen.

### **Analytische Modelle**

Der Ansatz in [166] nutzt ein analytisches Modell physikalischer Gesetze, um den Ladezustand zu prognostizieren. Hierfür werden zwar nur zwei Parameter benötigt,

allerdings müssen diese in umfangreichen Messungen mit realen Batterien bei unterschiedlichen Belastungen ermittelt werden. Ein ebenfalls auf Daten aus Messreihen basierender Ansatz wird in [167] gezeigt. Dieser ist für Sensorknoten mit Energiegewinnung aus der Umwelt gedacht und berücksichtigt Lade- und Entladeverhalten. Verglichen mit den anderen Modellen ist die Komplexität geringer, aber auch die Genauigkeit.

### 3.2.3 Messbasierte Ansätze

Messbasierte Ansätze nutzen observierbare Parameter einer Zelle, um auf den Ladezustand zu schließen. Der am einfachsten zu ermittelnde Parameter ist hierbei die Spannung, welche sich ohne zusätzliche Hardware erfassen lässt. Jeder Typ von Batterie und Akkumulator weist, wie in Kapitel 2.1.1 beschrieben, eine charakteristische Spannungskurve auf, mit deren Hilfe man Rückschlüsse auf die verbleibende Ladung ziehen kann. Zusätzlich kann bei dieser Art von Ansätzen die Umgebungstemperatur einbezogen werden.

#### Tabellenbasierte Ansätze

Eine einfache Möglichkeit, den Ladezustand zu bestimmen, ist die Nutzung aufgezeichneter Daten. Damit vergleicht man die gemessene Spannung. In [21] wird dieser Ansatz verfolgt. Eine Tabelle im System gibt Auskunft über den Ladezustand bei einem bestimmten Spannungswert. Nachteile sind der zusätzlich erforderliche Speicherplatz, fehlerhafte Vorhersagen und der Aufwand, denn diese Form basiert auf der Erstellung einer Spannungskurve. Da diese, wie in Abbildung 2.6 auf Seite 14 dargestellt, eine Phase von nahezu linearem und sehr geringem Spannungsabfall enthält, müssen die Daten sehr genau sein, da andernfalls nur eine schlechte Vorhersage möglich ist. Zusätzlich ist diese Kurve durch die nicht-linearen Effekte nur bedingt für andere Temperaturen und Lastverhalten aussagekräftig. Um diese Probleme zu kompensieren, wurden die Daten in [168] um den zu erwartenden Fehler ergänzt. So lassen sich wichtige Entscheidungen verschieben und der Einfluss der Ungenauigkeiten verringern.

#### Verbrauchserfassung

Wie bei manchen Co-Prozessoren, die nur die Lade- und Entladeaktivität observieren, ist es möglich, den Verbrauch rein in Software abzuschätzen und so mithilfe der erwarteten Batteriekapazität den Ladezustand abzuschätzen. In [169] geschieht dies abstrakt auf der Ebene ganzer Programmteile. Deren Ausführung wird gezählt und der Systemverbrauch anschließend durch vorher ermittelte (im Allgemeinen also gemessene) Energiekosten abgeschätzt. Die Zeit, in der die Hardwarekomponenten

aktiv sind, wird für TinyOS in [170] und für Contiki in [171] genutzt, um so den Verbrauch genauer zu erfassen. Basis sind auch hier gemessene Werte des Energieverbrauchs der jeweiligen Komponenten. Dieser Ansatz wird in [172] um Einflussfaktoren der Spannung erweitert, wie die Abhängigkeit bestimmter Komponenten und die Effizienz von Spannungsreglern. Die Genauigkeit eines solchen Verfahrens ist von der Genauigkeit der Abschätzung der in der Batterie enthaltenen Energiemenge abhängig, da keiner der auftretenden Effekte betrachtet wird.

### **Spannungsvorhersage**

In [23] wurden für die Vorhersage der Spannung die Begriffe zielorientiertes Energiemanagement und Spannungsbudget im Kontext der Sensornetze eingeführt. Der entwickelte Ansatz sieht das System und seine Teile als Regelungsschleife. Die Schleife des Batteriemanagements versucht, die tatsächliche Spannung einer durch stückweise Linearisierung berechneten Spannung anzugleichen. Dazu werden Schlafphasen in den Programmablauf eingefügt. Dabei erfolgt die Berechnung der Schlafphasen als Optimierungsproblem auf einem angeschlossenen PC. Mit dieser Herangehensweise wird zwar keine Aussage über den Ladezustand getroffen, allerdings wird das Erreichen des Lebenszeitziels durch Adaption der Anwendungsperiode ermöglicht, ohne dass der tatsächliche Verbrauch oder die Spannungskurve bekannt sein müssen.

### **Sonstige**

Eine Aussage über den nahen Entladeschluss einer Zelle wird in [173] mithilfe des Spannungsunterschiedes zwischen Belastung und Ruhe getroffen. Dieser steigt mit sinkender Restenergie. Informationen über den Ladezustand sind somit nur begrenzt möglich. Andere Ansätze könnten jedoch von einer genaueren Vorhersage des Entladeschlusses profitieren.

#### **3.2.4 Zusammenfassung der Ansätze**

Wie Tabelle 3.1 zeigt, haben alle Ansätze, die den Ladezustand ermitteln, Vor- und Nachteile. Die Bewertung erfolgt dabei nach einem mehrteiligen Maßstab: sehr gut (++), gut (+), neutral (o), schlecht (-) und sehr schlecht (--). Bei den Hardwarebasierten Ansätzen wirken die hohen Kosten und der Energieverbrauch nachteilig. Modellbasierte Ansätze leiden unter ihrer Komplexität und dem Aufwand für die Vorbereitung. Einzig messbasierte Ansätze sind leichtgewichtig genug, um für tief eingebettete Systeme geeignet zu sein. Insbesondere die Spannungsvorhersage mithilfe von Linearisierung scheint geeignet, obwohl sie keine Aussage über den exakten Ladezustand trifft, sondern nur eine Steuerung des Verbrauchs ermöglicht, sodass das Lebenszeitziel erreicht wird.

Tabelle 3.1: Bewertung der verschiedenen Ansätze zur Abschätzung des Ladezustands

	Smart Battery [151, 152]	MAX17043 [153]	Physikalische Modelle [155, 156]	Äquivalenz-Schaltungen [157, 158, 159]	Stochastische Ansätze [160, 161]	Neuronale Netze [162, 163]	Fuzzylogik [164, 165]	Analytische Modelle [166, 167]	Tabellen-basierte Ansätze [21, 168]	Verbrauchserfassung [169, 170, 171, 172]	Spannungsvorhersage [23]
Flexibilität	-	-	--	--	--	+	o	o	o	++	++
Genauigkeit	+	+	++	o	+	o	o	+	-	--	o
Kosten	--	-	++	++	++	++	++	++	++	++	++
Energieverbrauch	--	o	++	++	++	++	++	++	+	++	+
Vorbereitung	++	++	--	-	-	-	--	-	-	-	++
Laufzeitaufwand	+	++	--	-	o	o	o	-	++	++	+

### 3.3 Hinweis-basiertes Entlademanagement

Der im Folgenden vorgestellte Ansatz des Hinweis-basierten Entlademanagements ist von [23] inspiriert. Es nutzt auch den Spannungsabfall als Kontrollwert, verwendet aber weder explizite Regelungstechnik noch steuert es das Anwendungsverhalten direkt. Stattdessen nutzt es einfache Linearisierungen und ein Hinweissignal. Letzteres stellt eine Abstraktion dar, um einen geeigneten Energiemanager über eine etwaige notwendige Anpassung des Energieverbrauchs zu informieren.

Um die genannten Schwierigkeiten bei der Ermittlung der verfügbaren Energiemenge  $E(t)$  zu überwinden, nutzt der hier vorgestellte Ansatz die gemessene Spannung  $V_m(t)$  als Kontrollwert. Informationen über die aktuelle Spannung sind auf nahezu jeder Plattform verfügbar und erfordern wenig bis gar keinen Hardware-Mehraufwand. Trotz der eher schlechten Korrelation zwischen Spannung und verfügbarer Kapazität eignet sich der Spannungsabfall dennoch dazu, Informationen über den Ladezustand zu gewinnen.

Abbildung 3.4 zeigt das System als abstrakte Regelungsschleife, in der das Entlademanagement als Regelung fungiert. Ein signifikanter Unterschied zwischen angenommener Spannung  $V_{ref}$  und gemessener Spannung  $V_m$  führt zu einem Hinweis an den Energiemanager, der daraufhin das Energieprofil der Anwendung anpasst. Das führt zu einer Veränderung des Spannungsverlaufs, auf welche das System wiederum reagiert.

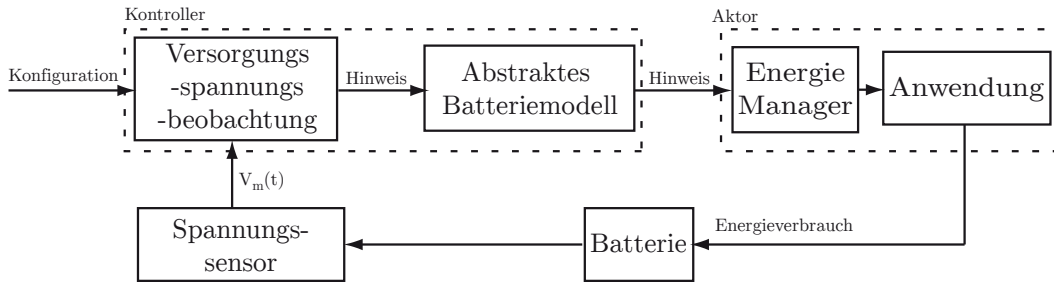


Abbildung 3.4: Abstrakte Regelungsschleife des Entlademanagements mit Nutzung einer Batterie als Energiequelle inklusive Energiemanagement und Anwendung

Das Batteriemanagement besteht aus zwei wesentlichen Teilen, der *Versorgungsspannungsbeobachtung* (VSB) und dem *abstrakten Batteriemodell* (ABM). Nur die VSB ist zwingend notwendig. Sie kann sowohl direkt einen Hinweis an den Energiemanager geben als auch dem ABM den gesamten internen Zustand zur Verfügung stellen. Dies dient zur Veränderung des Hinweises durch das ABM, wenn ein bestimmtes Verhalten der Batterie bekannt ist. Der Energiemanager, der den Energieverbrauch des Systems und der Anwendung aufgrund des Hinweises anpasst, ist nicht Teil des Batteriemanagements und kann unterschiedliche Charakteristiken besitzen, beispielsweise eine einfache Anpassung eines Arbeitszyklus oder ein komplexeres Verhalten umsetzen.

### 3.3.1 Versorgungsspannungsbeobachtung

Die VSB überwacht die gemessene Spannung  $V_m(t)$  und vergleicht sie mit der berechneten Referenzspannung  $V_{ref}(t)$  (siehe unten). Eine obere  $U(t)$  oder untere  $L(t)$  Schranke stellt den tolerierten Unterschied zwischen angenommener und gemessener Spannung dar. Werden diese überschritten, erzeugt die VSB einen Hinweis. Die Schranken werden wie in den Formeln 3.1 und 3.2 dargestellt als Anteil des verbleibenden Spannungsbudgets, also die verbleibende Spannung bis zur Entladeschlussspannung, berechnet. Dabei wird die Größe des Anteils durch vordefinierte Konstanten  $B_+$  und  $B_-$  festgelegt und es gilt  $V_{ref}(0) = V_m(0)$ .

$$U(t) = V_{ref}(t) + (V_{ref}(t) - V_{cutoff}) * B_+ \quad (3.1)$$

$$L(t) = V_{ref}(t) - (V_{ref}(t) - V_{cutoff}) * B_- \quad (3.2)$$



In jedem Zeitschritt wird ein Spannungsabfall um  $\Delta V(t)$  angenommen.  $\Delta V(t)$  berechnet sich aus dem verbleibenden Spannungsbudget und Zeitschritten (Formel 3.3).  $\Delta V(t)$  wird nur neu berechnet, wenn  $U(t)$  oder  $L(t)$  verletzt wurden.

$$\Delta V(t+1) = \begin{cases} \Delta V(t) & L(t) < V_m(t) < U(t) \\ \frac{V_m(t) - V_{cutoff}}{T - t} & \text{sonst} \end{cases} \quad (3.3)$$

$V_{ref}(t+1)$  repräsentiert die angenommene Spannung zum nächsten Zeitschritt. Wenn eine Schranke überschritten wurde, berechnet sie sich wie in Formel 3.4 beschrieben aus der gemessenen Spannung, andernfalls aus der angenommenen Spannung des aktuellen Zeitschrittes.

$$V_{ref}(t+1) = \begin{cases} V_{ref}(t) - \Delta V(t) & L(t) < V_m(t) < U(t) \\ V_m(t) - \Delta V(t) & \text{sonst} \end{cases} \quad (3.4)$$

Wenn  $U(t)$  oder  $L(t)$  verletzt wurden, muss der Energieverbrauch angepasst werden. Die VSB berechnet den Hinweis  $H(t)$ , um anzuzeigen, wie stark die Schranke verletzt wurde (siehe Formel 3.5). Der Hinweis wird dabei zum nächsten Ganzzahlwert gerundet, um die anwendungsseitige Reaktion zu vereinfachen.

$$H(t) = \begin{cases} -\frac{V_m(t) - V_{ref}(t)}{L(t) - V_{ref}(t)} & V_m(t) < L(t) \\ \frac{V_m(t) - V_{ref}(t)}{U(t) - V_{ref}(t)} & V_m(t) > U(t) \\ 0 & \text{sonst} \end{cases} \quad (3.5)$$

Das Verhalten des Systems ist beispielhaft in Abbildung 3.5 dargestellt. Im zweiten Zeitschritt sinkt die Spannung unter  $L(t)$ , woraufhin ein Hinweis generiert und  $\Delta V(t)$  angepasst wird. In dieser Darstellung führt der Hinweis zu einer Anpassung des Energieverbrauchs, woraufhin die Spannung im nächsten Zeitschritt weniger stark abfällt.

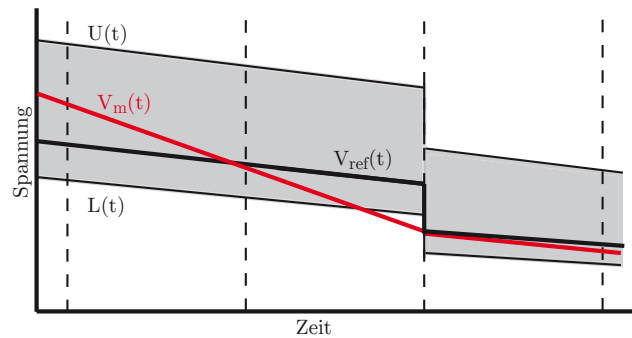


Abbildung 3.5: Systemverhalten mit durch Unterschreitung der Schranke ausgelöster Neuberechnung von  $\Delta V(t)$  und Anpassung des Energieverbrauchs

Notwendige Parameter für die Konfiguration der VSB sind die Konstanten  $B_+$  und  $B_-$ , die Länge und Zahl der Zeitschritte und die Entladeschlussspannung  $V_{cutoff}$ . Während letztere systemabhängig ist, definieren die Konstanten und die Anzahl der Zeitschritte die Reaktionsfähigkeit des Systems. Zur Laufzeit ist die VSB nur von  $V_m$  und  $t$  abhängig.

### 3.3.2 Abstraktes Batteriemodell

Wenn nur das Hinweissignal der VSB zur Anpassung des Verbrauchs genutzt wird, wird möglicherweise bei der Nutzung von Batterien oder Akkumulatoren nicht die gewünschte Anwendungsqualität erreicht. So würde der starke Spannungsabfall in der ersten Phase des Spannungsverlaufs (vgl. Phase A in Abbildung 2.6, Seite 14) zu einem negativen Hinweissignal in den ersten Zeitschritten führen, woraufhin weniger Arbeit vom System geleistet würde, bis positive Hinweise das negative wieder ausgleichen. Um die Qualität der Hinweise und somit die allgemeine Anwendungsqualität zu verbessern, ist es sinnvoll, einfache Informationen über das Batterieverhalten einzubeziehen.

Nimmt man ein Entladeverhalten wie in Abbildung 2.6 dargestellt an, passt das ABM durch einfache Regeln die Hinweise an dieses Verhalten an:

- Um negative Hinweissignale durch den starken Spannungsabfall in Phase A zu verhindern, wird dieser in den ersten Zeitschritten ignoriert.
- Zusätzlich ist es möglich, dass eine Batterie schon zu Beginn der Laufzeit nicht mehr die volle Kapazität besitzt. Wenn die initiale Spannung unter der zur erwartenden Spannung liegt, kann das ABM ein entsprechendes Hinweissignal generieren.
- Wenn das verbleibende Spannungsbudget nur noch klein ist (Phase C), werden negative Hinweise verstärkt, wohingegen positive verringert werden. So kann sichergestellt werden, dass die Spannung nicht unter die Entladeschlussspannung fällt. In den letzten Zeitschritten werden positive Hinweise gänzlich ignoriert, da ein erhöhter Verbrauch in dieser Phase zu einem sehr schnellen Spannungsabfall führen kann.
- Verstärkt sich in den letzten Zeitschritten der Spannungsabfall, können negative Hinweise gegeben werden.
- Im Laufe der Entwicklung durchgeführte Experimente haben gezeigt, dass negative Hinweise der VSB im Falle von Störungen, z. B. aufgrund sinkender Temperatur, nur sehr langsam wieder ausgeglichen werden. Dies kann im Fall von sich wiederholenden Störungen zu einer kontinuierlichen Verringerung des

Energieverbrauchs führen. Um die Reaktivität zu erhöhen, werden die Hinweise aufsummiert. Änderungen in der Richtung des Hinweissignals werden entsprechend verstärkt und führen zu einer schnelleren Konvergenz in Richtung des Verbrauchs, der ursprünglich bei  $t_0$  festgelegt wurde.

- Da nach einem negativen Hinweis der Erholungseffekt zu einer steigenden Spannung führen kann, würde im nächsten Zeitschritt wahrscheinlich ein positiver Hinweis erzeugt. Um eine insbesondere bei Knopfzellen beobachtete Oszillation zu verhindern, wird ein positiver Hinweis verringert, wenn er direkt im Anschluss an einen negativen Hinweis auftritt.

Die Erkennung der Phasen ist vom Verhalten der Zelle abhängig und kann auf verschiedene Weise erfolgen. Neben fixen Zeit- oder Spannungsgrenzen (vergleiche tabellenbasierte Ansätze) kann bei Zellen mit charakteristischer Entladekurve (vergleiche Abbildung 2.6), beispielsweise Alkaline, eine Veränderung im Spannungsabfall als Phasenwechselpunkt genutzt werden. Ein Abflachen des Abfalls in Phase A zeigt den Übergang zu Phase B, wohingegen ein steigender Abfall einen Wechsel in Phase C andeutet.

### 3.4 Evaluation

Das vorgestellte System wurde in C++ für das Betriebssystem REFLEX [62] implementiert. Die Zielplattform war der eZ430-Chronos von Texas Instruments [144]. Es wurden verschiedene Szenarien entworfen, um die verschiedenen Bestandteile und deren Verhalten zu evaluieren. Die verwendete Anwendung entspricht einem vereinfachten Sense&Send-Verhalten von Sensornetzen. Periodisch werden Funknachrichten mit Statusinformationen an einen PC gesendet und der Rest der Zeit in einem energiesparenden Schlafzustand verbracht. Der integrierte Analog-Digital-Wandler des Systems wurde zur Erfassung der Batteriespannung genutzt. Die Reaktion der Anwendung auf etwaige Hinweissignale erfolgte durch Anpassen des Anwendungsintervalls.

Für die Experimente wurden verschiedene Parameter und Laufzeiten genutzt. Die Parameter wurden empirisch durch Experimente (siehe Experiment 1) bestimmt und abhängig von der Energiequelle ausgewählt. Da Experimente mit realistischem Energieverbrauch und somit Laufzeiten von mehreren Monaten nicht praktikabel sind, wurden abhängig von der Energiequelle möglichst kleine Anwendungsintervalle gewählt. Der resultierende Verbrauch verkürzte die Dauer der Experimente entsprechend.

Die Parameter des VSB und des ABM wurden mit dem Ziel gewählt, das Lebenszeitziel zu erreichen und möglichst die verfügbare Energie auszunutzen. Letzteres ist allerdings nachrangig. Dies schlägt sich insbesondere in der Dimensionierung

der Faktoren  $B_+$  und  $B_-$  zur Berechnung der Schranken nieder. Während diese für Energiespeicher mit linearem Spannungsabfall wie Kondensatoren gleich und je nach Speichertyp auch sehr klein sein können, müssen sie an das nicht lineare Verhalten von Batterien und Akkumulatoren angepasst werden. Aus diesem Grund wurden die Faktoren für die Berechnung der Schranken für alle Experimente mit Batterien als Energiequelle auf  $B_+ = 0,5$  und  $B_- = 0,1$  eingestellt, womit  $U(t)$  höher als  $L(t)$  war. Eine schnelle Reaktion auf eine abfallende Spannung ist essenziell, um  $T$  zu erreichen, während die Ausnutzung von potenziell ungenutzter Energie und somit eine verbesserte Anwendungsqualität durch positive Hinweise optional ist. Ein zu kleines  $B_+$  könnte dazu führen, dass der Verbrauch zu stark erhöht würde, was die Batterie weitgehend erschöpfen könnte. Dies könnte nur durch ein starkes negatives Hinweissignal kompensiert werden, was zu einer geringeren Anwendungsqualität führen würde, je weiter sich das System dem Lebenszeitziel nähert.

#### 3.4.1 Experiment 1 - Wirksamkeit der Versorgungsspannungsbeobachtung

Ziel des ersten Experimentes war die Evaluation der Parameter der VBS hinsichtlich der Reaktivität und der Fähigkeit, das Lebenszeitziel zu erreichen. Hierzu wurden sowohl die Größe der Schranken als auch die Stärke der Reaktion der angeschlossenen Anwendung variiert. Das abstrakte Batteriemodell wurde nicht genutzt, da im Versuchsaufbau eine lineare Energiequelle mit unterschiedlicher Kapazität simuliert wurde.

##### Aufbau

Während der Sensorknoten mit einer fixen Spannung versorgt wurde, stellte die simulierte Energiequelle auf Grundlage des erfassten Verbrauchs der VBS eine Spannung zwischen 3,2V und 2,2V zur Steuerung zur Verfügung, welche der verbleibenden Energiemenge entsprach. Der Spannungsbereich entsprach dem typischen Arbeitsbereich des verwendeten Knotens. Die Ausgangsladung variierte je nach Szenario zwischen 12,5mA und 37,5mA, wobei das System jedes Mal 25mA annahm. Die Anwendung reagierte auf die Hinweise der VSB durch Anpassung des Anwendungsintervalls. Der Regelbereich reichte von 3,125% bis 25%<sup>4</sup> des aktuellen Anwendungsintervalls. Durch den linearen Spannungsabfall der simulierten Energiequelle waren beide Schranken gleich groß.  $B_+$  und  $B_-$  variierten je nach Durchlauf zwischen 0,05 und 0,5. Die Überwachung durch die VSB und die Steuerung der Anwendung erfolgten in 50 Intervallen von je 30 Sekunden.

---

<sup>4</sup>Prozentuale Angaben basieren auf der Verwendung von leicht zu realisierenden Shiftoperationen.

## Ergebnisse

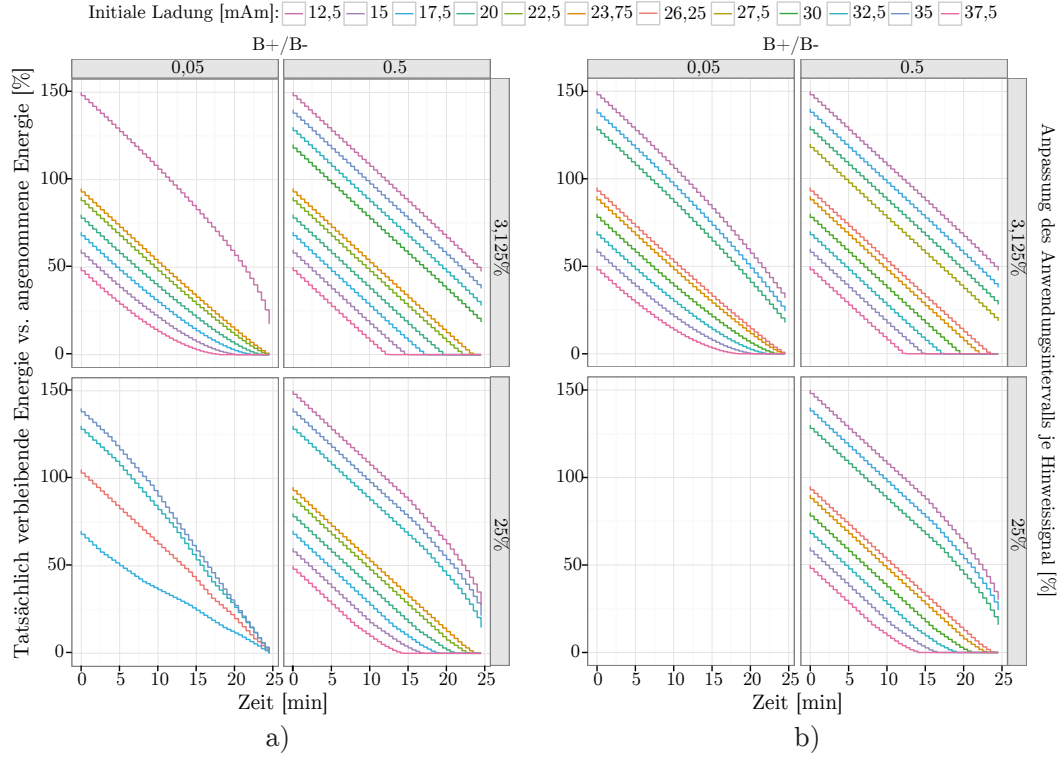


Abbildung 3.6: a) Verlauf der verbleibenden Energie unter verschiedenen Parameterkombinationen und initialen Ladungen: Es werden nur Fälle dargestellt, in denen das Lebenszeitziel nicht erreicht wurde oder mehr als 10% der Energie ungenutzt blieb. b) Die kleinstmögliche obere Schranke wurde auf den Spannungsabfall eines Intervalls limitiert.

Abbildung 3.6 a) zeigt eine Übersicht des Verlaufs der tatsächlich verfügbaren Energie unter den vier äußeren Parameterkonstellationen. Die vollständigen Ergebnisse der insgesamt 16 Kombinationen aus Regelbereich und Schranke sind in Abbildung A.1 im Anhang dargestellt. Es werden in den Abbildungen nur die Szenarien dargestellt, in denen das Lebenszeitziel nicht erreicht wurde oder mehr als 10% der Ausgangsladung ungenutzt blieben. Dies reflektiert die beiden Ziele: das Erreichen des Lebenszeitziels selbst und das möglichst restlose Ausnutzen der verfügbaren Energie.

Erfolgt nur eine geringe Reaktion auf die Hinweissignale und/oder reagiert das System durch hohe Schranken nur langsam, kann das Lebenszeitziel selbst bei Szenarien mit geringen Abweichungen nicht erreicht werden. Mit kleiner werdenden Schranken und steigender Reaktion kann dagegen in immer mehr Szenarien das Ziel erreicht werden. Wie sich der Abbildung entnehmen lässt, garantieren jedoch selbst eine niedrige Schranke und starke Reaktion nicht das Erreichen des Ziels. Dies betrifft

nicht nur Szenarien, in denen die angenommene Energie stark von der tatsächlichen abweicht. Ursache hierfür sind die Schranken, die in den letzten Zeitschritten sehr kleine Werte einnehmen. Selbst geringe Abweichungen der gemessenen Spannung von der erwarteten Spannung können dann zu hohen Hinweissignalen und entsprechender Anpassung der Energieverfügbarkeit führen. Während sich solche Überreaktionen im Falle negativer Hinweise zwar auf die Anwendungsqualität auswirken, aber unproblematisch für das Lebenszeitziel sind, können positive Hinweise die angenommene Energie deutlich über die tatsächlich verfügbare Energie hinaus ansteigen lassen und somit das Lebenszeitziel gefährden. In den letzten Zeitschritten kann in einem solchen Fall auch nicht mehr gegengesteuert werden. Die Regeln des abstrakten Batteriemodells sollen ein solches Verhalten vermindern. Da dieses in diesem Experiment jedoch nicht genutzt wurde, kann dem Problem auf zwei Arten begegnet werden: Zum einen können Sicherheitspuffer in Form zusätzlicher Laufzeit hinzugefügt werden. Bereits ein zusätzliches Intervall ermöglichte es auch in den Grenzfällen, das Lebenszeitziel zu erreichen. Zum anderen kann die obere Schranke so verändert werden, dass kleine Abweichungen nicht mehr zu großen Hinweissignalen führen.

Abbildung 3.6 b) und Abbildung A.2 im Anhang stellen die Ergebnisse mit modifizierter oberer Schranke dar. Diese wurde so angepasst, dass der kleinstmögliche Wert dem Spannungsabfall  $\Delta V(t)$  innerhalb eines Intervalls entsprach. Je weniger Intervalle verblieben, desto größer wurde die Schranke im Vergleich zum verbleibenden Spannungsabfall, wodurch nur entsprechend große Abweichungen zu einem Hinweissignal führten. Wie die Abbildung zeigt, wurde das oben genannte Problem wirksam unterdrückt. Die Kombination von niedrigen Schranken und starker Reaktion war in der Lage, selbst in extremen Szenarien das Lebenszeitziel zu erreichen. Jedoch stieg die Zahl der Szenarien, in denen mehr als 10% der Energie ungenutzt blieben, da positive Hinweise durch die Modifikation verhindert wurden.

Das Experiment demonstriert die grundsätzliche Wirksamkeit des Ansatzes. Abweichungen der angenommenen Energie von der tatsächlich vorhandenen Energie können in Abhängigkeit von den verwendeten Parametern durch eine Reaktion auf die Hinweise ausgeglichen werden, sodass Lebenszeitziele erreicht werden. Eine hohe Reaktivität und ausreichend starke Reaktionen ermöglichen es, selbst große Abweichungen auszugleichen, wobei Überreaktionen jedoch verhindert werden müssen.

### **3.4.2 Experiment 2 - Versorgungsspannungsbeobachtung für Superkondensatoren**

Ziel dieses Experimentes war die Evaluation der Versorgungsspannungsbeobachtung bei Nutzung von Superkondensatoren. Analog zum ersten Experiment wurde für diese Evaluation das abstrakte Batteriemodell nicht genutzt. In mehreren Durchläufen

wurde die angenommene Kapazität der aus einem Superkondensator bestehenden Energiequelle variiert.

### Aufbau

Als Energiequelle wurde ein 1F Superkondensator eingesetzt und jeweils von 3,2V auf 2,2V entladen. Die vom System angenommene Kapazität variierte zwischen 0,25F und 4F. Die Parameter des Systems wurden auf Grundlage des ersten Experimentes ausgewählt. Die Anwendung reagierte auf die Hinweise der VSB mit einer Anpassung des Intervalls um 12,5%.

Da Kondensatoren einen nahezu linearen Spannungsabfall aufweisen, wurden bei-  
de Schranken  $B_+$  und  $B_-$  auf 0,05 festgelegt. Die Überwachung durch die VSB und die Steuerung des Anwendungsintervalls erfolgten in 50 Intervallen von je 30 Sekunden.

### Ergebnisse

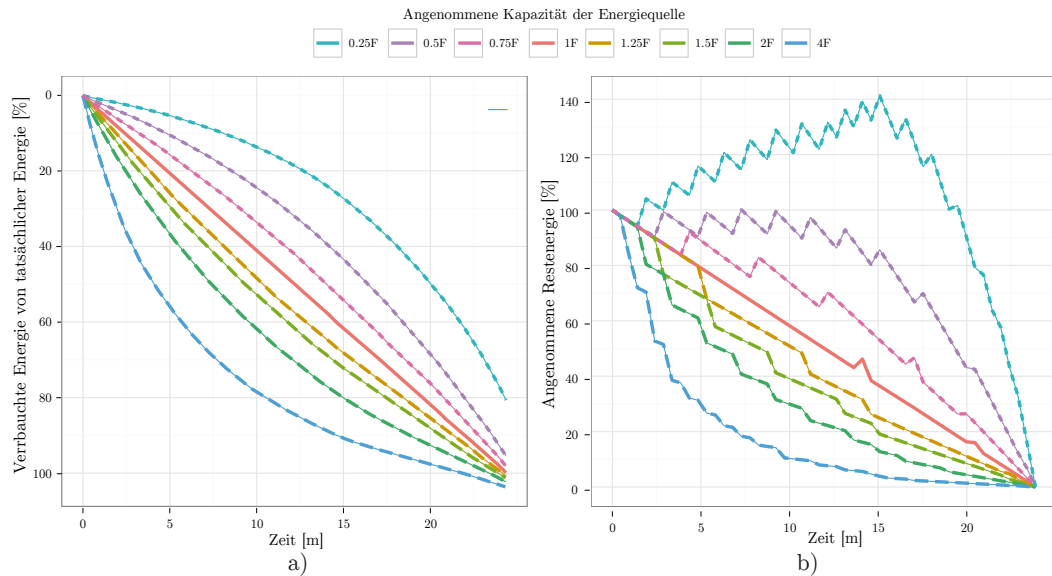


Abbildung 3.7: Management-Experiment mit einem 1F Superkondensator als Energiequelle: Die angenommene Ladungsmenge variiert zwischen 0,25F und 4F. a) prozentualer Verbrauch der tatsächlichen Energie, b) Angenommene Restenergie

Abbildung 3.7 stellt die Ergebnisse des Experiments grafisch dar, dabei zeigt Abbildung 3.7 a) den Prozentsatz, der von den tatsächlich verfügbaren 1F bereits verbraucht wurde. Abbildung 3.7 b) stellt die Sicht des Systems auf die verfügbare Energie und die aus den Hinweisen resultierende Anpassung dar.

Selbst wenn die angenommene Kapazität vier mal so groß wie die tatsächliche ist, ist die VSB in der Lage, den stärkeren Spannungsabfall zu erkennen und entspre-



chende Hinweissignale zu erzeugen. Die Auswirkungen der Hinweise lassen sich gut in Abbildung 3.7b erkennen, da sich die angenommene Restenergie dort mit jedem Hinweis sichtbar verändert. Auch im umgekehrten Fall, in dem die zur Verfügung stehende Energie größer ist als angenommen, reagiert die VSB mit dem entsprechenden Hinweissignal auf den in diesem Fall unerwartet geringen Spannungsabfall. Allerdings wird nicht der komplette Unterschied zur realen Energiemenge ausgeglichen.

Insgesamt zeigt das Experiment, dass der mit der VSB verfolgte Ansatz zumindest für die linearen Superkondensatoren geeignet ist, auf Variationen in der zur Verfügung stehenden Energiemenge zu reagieren und einen Ausgleich zu ermöglichen.

### 3.4.3 Experiment 3 - abstraktes Batteriemodell

Ziel dieses Experimentes war es, die Endladeüberwachung und ihre Hinweissignale dem nichtlinearen Endladeverhalten von konventionellen Zellen auszusetzen.

#### Aufbau

Für dieses Experiment wurden zwei in Reihe geschaltete Eneloop light AAA Akkumulatoren [174] genutzt, die das System direkt versorgten. Sie wurden ausgewählt, da die angegebene Kapazität mit 550mAh deutlich kleiner als die von Batterien ist und somit die Laufzeit des Experimentes verringert werden konnte.

Das Lebenszeitziel wurde als Kompromiss aus Laufzeit und Last auf 72 Stunden festgelegt und in  $T=160$  Zeitschritte eingeteilt. Die systembedingte Endladeschlussspannung betrug 2,2V.

Wie bereits motiviert, müssen die Schranken der VSB an das nicht lineare Verhalten von Batterien und Akkumulatoren angepasst werden. Um das gewünschte Verhalten zu realisieren, wurden  $B_+ = 0,5$  und  $B_- = 0,1$  festgelegt. Diese Werte haben sich in Experimenten als geeigneter Kompromiss zwischen Reaktionsschnelligkeit bei zu großem Spannungsabfall und konservativer Erhöhung des Verbrauchs erwiesen.

Die durch den Ansatz erzeugten Hinweissignale wurden vom System genutzt, um das Anwendungsintervall um 12,5% je Hinweispunkt anzupassen. Während der Phase A wurden negative Hinweissignale vom ABM unterdrückt, da diese aufgrund des Spannungsabfalls in dieser Phase erzeugt werden. Der Wechsel zur Phase B erfolgte, sobald der Unterschied zwischen der gemessenen  $V_m$  und der berechneten Spannung  $V_{ref}$  geringer wurde, was eine Stabilisierung der Spannung anzeigte. Phase C begann ab einer verbleibenden Spannung von 2,4V. Das ABM unterdrückte positive Hinweissignale während Phase C, da sich zu diesem Zeitpunkt der Spannungsabfall verstärkt auswirkt.



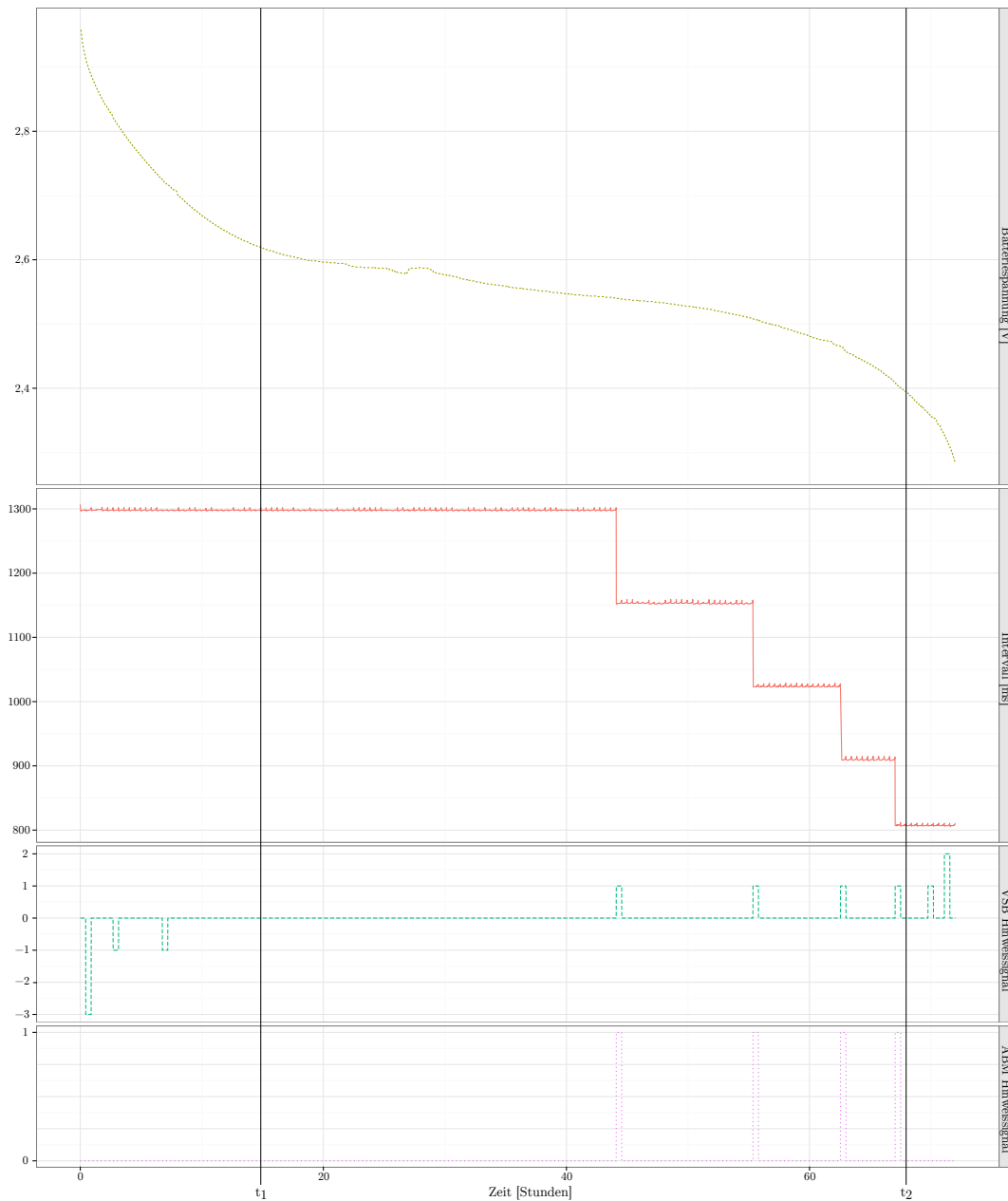


Abbildung 3.8: Management-Experiment über 72 Stunden mit Eneloo light 550mAh Akkumulator: Das Anwendungsintervall wurde durch Hinweissignale berechnet. Ein kleineres Intervall verursacht einen höheren Verbrauch.

## Ergebnisse

Das Ergebnis des Experimentes ist in Abbildung 3.8 dargestellt. Das Lebenszeitziel wurde nicht nur erreicht, sondern im Vergleich mit einem festgelegten Anwendungsintervall wurde auch mehr Arbeit vom System geleistet. Da die Kapazität größer war als angenommen, wurden mehrfach positive Hinweise generiert. Das Anwendungsintervall wurde dadurch nach und nach geringer, wodurch 11,6% mehr Arbeit geleistet wurde als mit einem fixen Intervall.

Das abstrakte Batteriemodell arbeitet wie gewünscht und gleicht Hinweise der Versorgungsspannungsbeobachtung bei Bedarf aus. So werden die negativen Hinweise am Anfang des Experimentes bis zum Beginn der Phase B bei  $t_1$  nicht an den Energiemanager weitergereicht. Mit Erreichen von Phase C bei  $t_2$  wurden positive Hinweise ebenso unterdrückt, was dazu führte, dass bei Erreichen von  $T$  die Spannung mit 2,28V nicht der Entladeschlussspannung von 2,2V entsprach und somit potenziell etwas mehr Arbeit hätte geleistet werden können.

Das Ergebnis zeigt die grundsätzliche Fähigkeit des abstrakten Batteriemodells zur Modifikation der Hinweise der Versorgungsspannungsbeobachtung, so dass ein Erreichen des Lebenszeitziels auch bei einem nichtlinearen Spannungsverlauf und Entladeverhalten von Batterien möglich ist.

### 3.4.4 Experiment 4 - verringerte Kapazität

In diesem Experiment wurde das Verhalten des Systems evaluiert, wenn die initiale Ladung geringer ist als von der Batterietechnologie zu erwarten ist. Zu diesem Zweck wurde die Spannung zu Beginn der Laufzeit mit der zu erwartenden verglichen. Bei einer Abweichung reagiert das ABM mit negativen Hinweisen. Das Experiment wurde mit dem vorgestellten Ansatz und zusätzlich mithilfe eines Ladezustandsmessers MAX17043 [153] durchgeführt.

## Aufbau

Als Energiequelle wurde ein 40 mAh Li-Ion-Akkumulator eingesetzt, dessen Spannung durch einen linearen Spannungsregler auf 3,0V abgesenkt wurde. Es wurden zwei Durchläufe durchgeführt. Im ersten (HW) wurde der durch den MAX17043 ermittelte Ladezustand genutzt, um die verbleibende Energiemenge zu bestimmen und daraus das Anwendungsintervall in jedem Zeitschritt zu berechnen. Der zweite Durchlauf (SW) nutzt die durch den vorgestellten Ansatz erzeugten Hinweissignale, um das Anwendungsintervall gegebenenfalls um 12,5% je Hinweispunkt anzupassen.

In diesem einfachen Experiment beruht das Hinweissignal zum Systemstart auf dem initialen Spannungsbudget. Dieses entspricht der Differenz zwischen der Spannung zu Beginn der Laufzeit und der Entladeschlussspannung. Weicht das Span-

nungsbudget um mehr als 12,5% von dem durch das ABM angenommenen Budget ab, wird jeweils ein negativer Hinweis generiert. Die 12,5% sind in der Reaktion der Anwendung begründet, die mit jedem Hinweis ihr Intervall um 12,5% anpasst. Die Parameter des Systems entsprechen denen des vorangegangenen Experimentes, jedoch startet das Experiment mit einer nur 80%-ig geladenen Zelle.

## Ergebnisse

Abbildung 3.9 zeigt neben dem durch den MAX17043 ermittelten Ladezustand die von VSB und ABS erzeugten Hinweise sowie die daraus resultierende Anwendungsfrequenz.

In beiden Durchläufen wurde das Lebenszeitziel erreicht. Während die Anwendungsfrequenz beim HW-Durchlauf analog zum Ladezustand schwankt, ist der SW-Ansatz stabil und reagiert aufgrund der hohen Schranken erst spät. Die Schwankungen des gemessenen Ladezustandes bzw. Abweichung vom idealen Ladezustandsverlauf sind Ursache der deutlichen Erhöhung des Anwendungsintervalls zum Ende der Laufzeit. Dies liegt in der direkten Verbindung von gemessenem Ladezustand und verbleibender Energiemenge und sollte sich mit komplexeren Übertragungsfunktionen stabilisieren lassen.

Durch die beim HW-Durchlauf höhere Anfangsfrequenz ist der Ladezustand durchgängig niedriger als beim SW-Durchlauf. Die Anzahl an Anwendungsausführungen ist nahezu identisch; beim SW-Durchlauf werden geringfügig (0,7%) mehr Funknachrichten versendet.

Durch die von der Erwartung für diesen Zellentyp abweichende Spannung zu Beginn (3,99V statt 4,2V) wurde beim SW-Durchlauf durch das ABM ein negativer Hinweis erzeugt, welcher die zur Verfügung stehende Energiemenge und somit die Anwendungsfrequenz verringerte. Analog zu den vorangegangenen Experimenten wurden durch die VSB im weiteren Verlauf positive Hinweise generiert, was die Anwendungsfrequenz erhöhte. Nach Erreichen von Phase C ( $T_1$ ) wurden positive Hinweise durch das ABM unterdrückt. In den letzten beiden Zeitschritten wurden aufgrund des steigenden Spannungsabfalls negative Hinweise durch das ABM erzeugt.

Das Experiment zeigt eine nahezu identische Leistungsfähigkeit von Software-basiertem und Hardware-basiertem Ansatz ohne Mehraufwand durch Hardwarekosten. Durch die negativen Hinweise am Anfang der Laufzeit ist es möglich, das Lebenszeitziel zu erreichen, ohne dass starke Fluktuationen in der zur Verfügung stehenden Energiemenge erzeugt werden.

### KAPITEL 3. LADEZUSTANDSÜBERWACHUNG ZUM ERREICHEN VON LEBENSZEITZIELEN

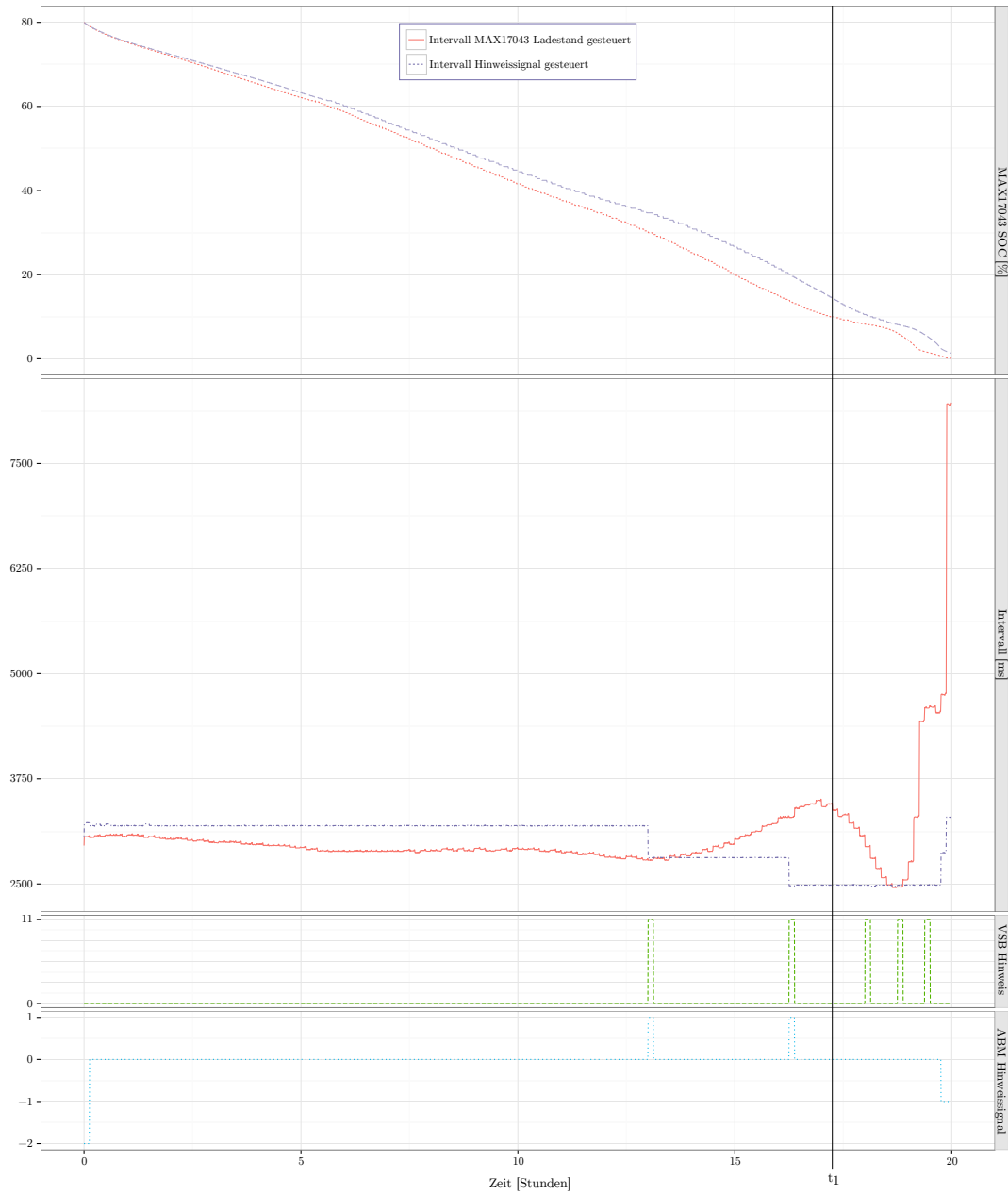


Abbildung 3.9: Management-Experiment über 20 Stunden mit zu 80% geladenem 40mAh Li-Ion-Akkumulator: Das Intervall wurde durch Hinweissignal oder MAX17043-Ladezustand berechnet. Ein kleineres Intervall verursacht einen höheren Verbrauch.

### 3.4.5 Experiment 5 - temperaturbedingte Störungen

In diesem Experiment wurde überprüft, inwieweit das System in der Lage ist, auf temperaturbedingte Störungen zu reagieren. Dazu wurden die Knoten mehrere Perioden lang einer verringerten Temperatur ausgesetzt. Es wurden sowohl CR2032 Knopfzellen [175] als auch in Reihe geschaltete handelsübliche No-Name Alkaline AAA Batterien genutzt. Zusätzlich wurde bei jedem Experiment ein weiterer Knoten mit einem konstanten Anwendungsintervall verwendet, um die Auswirkungen der Störungen aufzuzeigen.

#### Aufbau

Die Parameter beider Experimente sind in Tabelle 3.2 dargestellt. Der grundsätzliche Aufbau ähnelt sich, allerdings wurden aufgrund der unterschiedlichen Eigenschaften der Zellen verschiedene Temperaturen, Lebenszeitziele und Anwendungsfrequenzen genutzt.

Tabelle 3.2: Parameter des 5. Experiments

	Duracell	AAA Alkaline
T	80 je 12 Minuten	130 je 1 Stunde
initiales Intervall	1s	0,6s
normale Temperatur	$\approx 19^\circ\text{C}$	$\approx 6^\circ\text{C}$
tiefe Temperatur/Anzahl/Dauer	$\approx 6^\circ\text{C}/2/3$ Stunden	$\approx -22^\circ\text{C}/5/8$ Stunden

Ausgehend von im Vorfeld bei Raumtemperatur aufgezeichneten Spannungskurven wurden die Zeitschritte für Phase A auf 10% von  $T$  festgelegt und Phase C begann bei 20% des verbleibenden Spannungsbudgets. Das Anwendungsintervall wurde mit jedem Hinweis-Punkt um 10% erhöht oder verringert.

#### Ergebnisse

Das Ergebnis des Experimentes mit Knopfzellen zur Energieversorgung ist in Abbildung 3.10 dargestellt. Der Spannungsabfall aufgrund der niedrigeren Temperatur bei  $t_0$  und  $t_1$  ist deutlich zu erkennen. Der Referenzknoten, der ein stabiles Intervall von einer Sekunde nutzte, unterschritt ab  $t_2$  die Entladeschlussspannung von 2,2V und begann anschließend durch den Spannungsabfall neu zu starten. Als Folge lieferte dieser Knoten nur rund 67,3% der möglichen Nachrichten. Der Knoten, der sein Intervall mit dem präsentierten Ansatz anpasste, erreichte  $T$ , lieferte aber rund 10,8% weniger Nachrichten aus, als bei voller Energie und einem fixen Intervall theoretisch möglich wären.

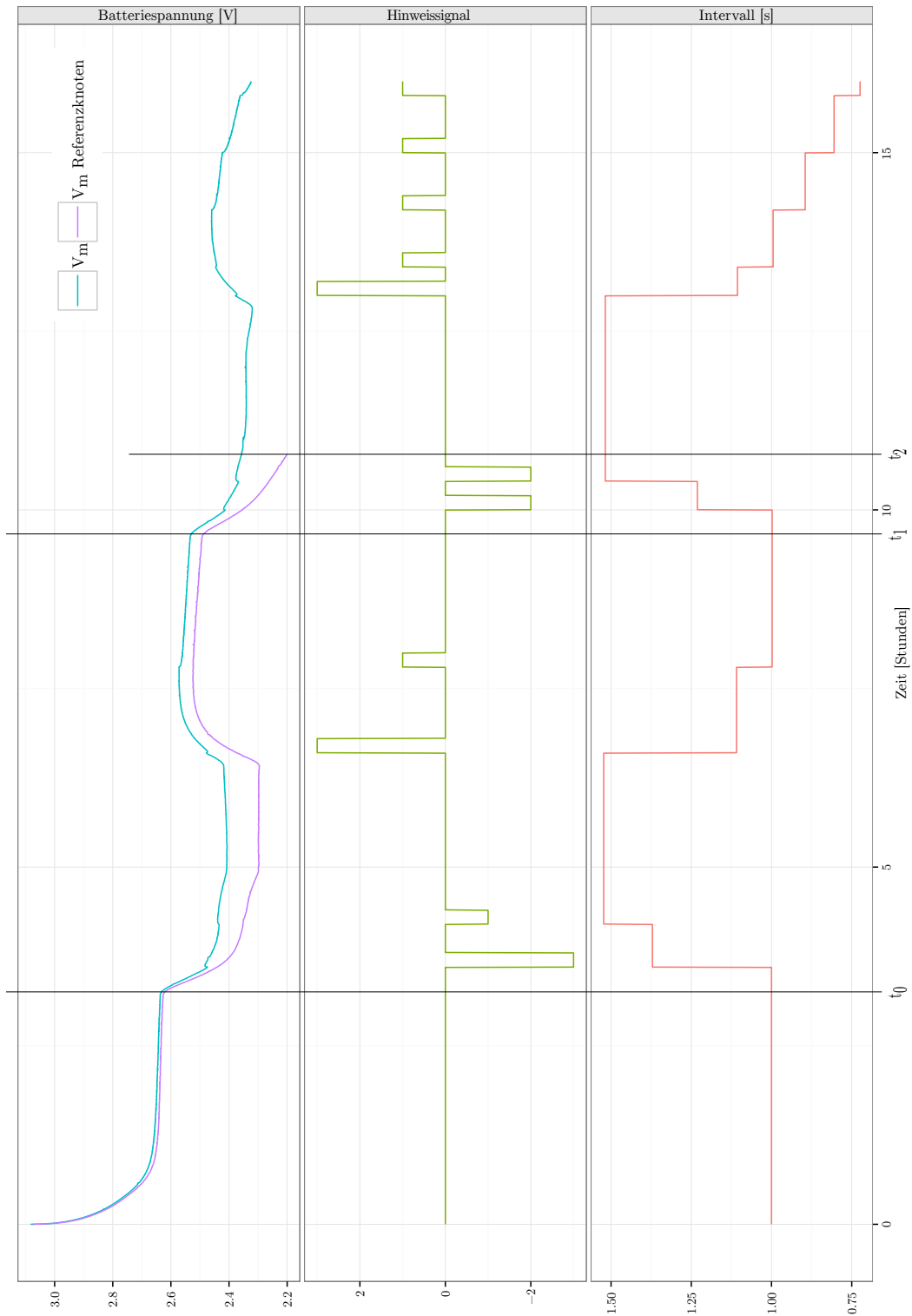


Abbildung 3.10: Management-Experiment über 16 Stunden mit CR2032 Knopfzelle:  
Ein kleineres Intervall verursacht einen höheren Verbrauch.

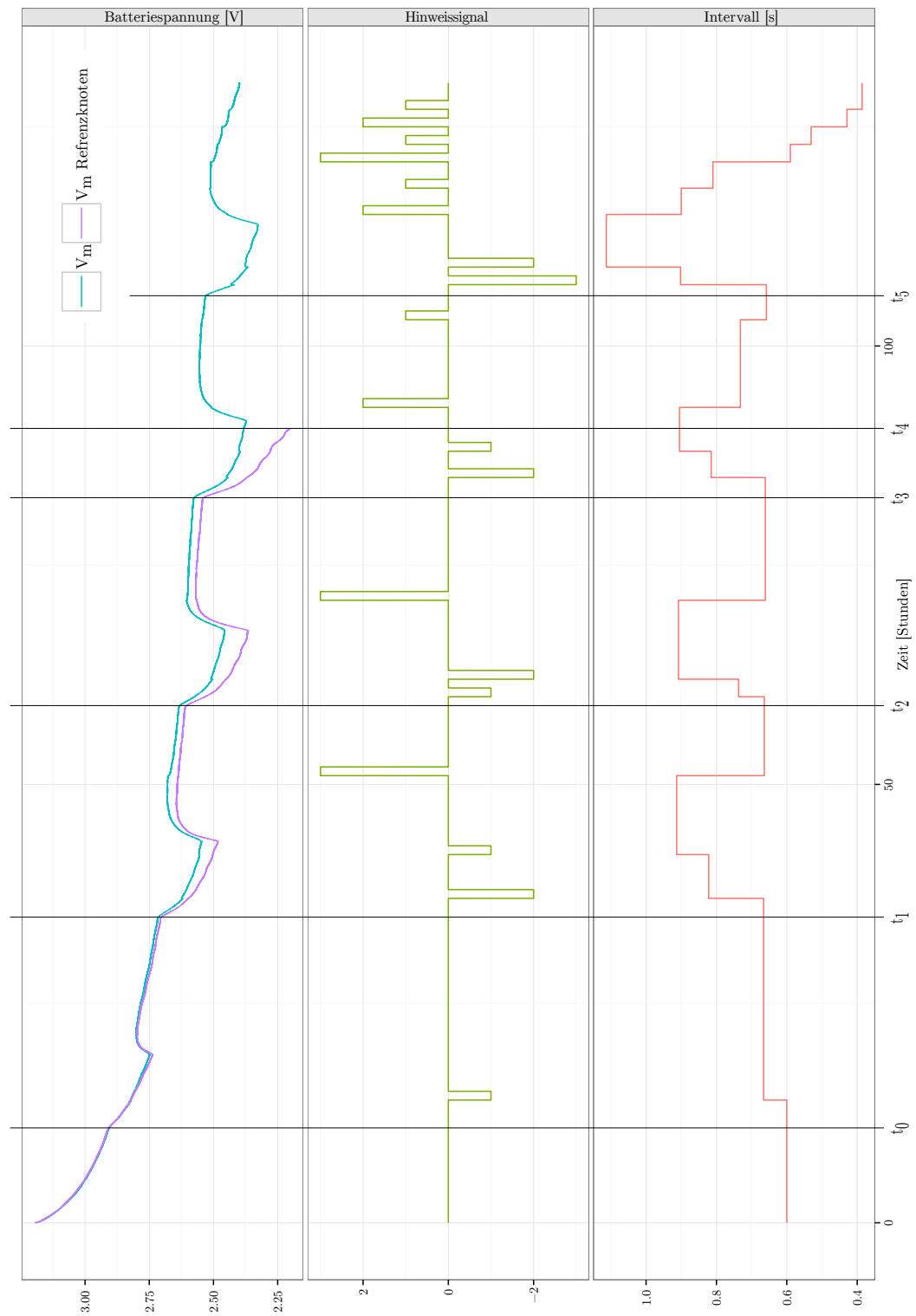


Abbildung 3.11: Management-Experiment über 130 Stunden mit AAA Batterien: Je kleiner das Intervall desto größer der Verbrauch.

Wie in der Abbildung dargestellt ist, wurde bedingt durch die Regeln des ABM während des starken Spannungsabfalls am Anfang des Experimentes kein negativer Hinweis erzeugt. In der ersten Phase mit niedriger Temperatur beginnend bei  $t_0$  wurde  $L(t)$  unterschritten und als Folge des negativen Hinweises das Intervall vergrößert. Nach dieser ersten Phase wurde mehrmals  $U(t)$  überschritten und positive Hinweise gegeben, was das Ursprungsintervall wieder herstellte.

Nach dem Beginn der zweiten Niedrigtemperaturphase bei  $t_1$  reagierte das System erneut auf den Spannungsabfall und erzeugte wiederum mehrfach negative Hinweissignale.

Nachdem sich die Spannung wieder stabilisierte, wurde das Intervall mehrfach verkleinert. Gegen Ende des Experimentes wurde  $U(t)$  mehrfach überschritten und das Anwendungsintervall dementsprechend verringert. Trotzdem reichte dies nicht aus, um das Spannungsziel von 2,2V zu erreichen. Allerdings konnte erfolgreich auf die temperaturbedingten Spannungseinbrüche reagiert und so, im Gegensatz zu einem festen Intervall, das Lebenszeitziel erreicht werden.

Abbildung 3.11 stellt das Ergebnis des Experimentes mit AAA Alkaline Zellen zur Energieversorgung dar. Beide Knoten verhalten sich ähnlich wie im vorhergehenden Experiment mit Knopfzellen. Die Niedrigtemperaturphasen führen zu einem Spannungsabfall, der einen negativen Hinweis auslöst und so das Intervall der Applikation vergrößert. Damit wird der Energieverbrauch verringert. Die Auswirkung lässt sich im Vergleich mit der Spannung des Referenzknotens erkennen. Dessen Spannung weicht ab  $t_1$  deutlich erkennbar von  $V_m$  ab und erholt sich auch nach den Niedrigtemperaturphasen nicht vollständig. Das führte zu einem Ausfall des Referenzknotens bei  $t_4$ .

Ab  $t_1$  wird nach einer Phase mit niedriger Temperatur das erzeugte negative Hinweissignal wieder ausgeglichen. Gegen Ende der geplanten Lebenszeit werden kontinuierlich positive Hinweise erzeugt, da das Spannungsbudget dies zulässt. Das Spannungsziel wurde nicht erreicht und es wurden ca. 12,88% weniger Nachrichten versendet als möglich, aber der Knoten konnte bis zum Lebenszeitziel überleben. Im Gegensatz dazu konnte der Referenzknoten das Lebenszeitziel nicht erreichen und dadurch nur rund 71% der Daten übertragen.

Experiment 5 zeigt, wie groß der Einfluss der Temperatur auf die Spannung ist. Nur wenn auf solche Veränderungen der Spannung reagiert wird, ist ein Erreichen des Lebenszeitziels möglich. Zukünftig könnte eine Kopplung der Hinweissignale mit der gemessenen Temperatur bei einer schärferen Einstellung der Schranken eine Verbesserung der Energieausnutzung ermöglichen.



### 3.5 Zusammenfassung

In diesem Kapitel wurde ein Ansatz für eine Entladeüberwachung vorgestellt, der komplexe Berechnungen zur Ermittlung des Ladezustandes von Batterien vermeidet. Dabei wird der Spannungsabfall genutzt, um der Anwendung Hinweise zu geben, ob diese den Energieverbrauch senken muss oder erhöhen kann. Die Versorgungsspannungsbeobachtung ist nur auf wenige Parameter angewiesen, deren Werte von der gewünschten Laufzeit und der verwendeten Art der Energiequelle abhängen. Die Parameter können technologieübergreifend für unterschiedliche Batterietypen verwendet werden. Das abstrakte Batteriemodell ermöglicht eine universelle Verwendbarkeit, ohne dass komplexe Parameter benötigt werden oder langwierige Kalibrierungen erfolgen müssen.

Zusätzlich ermöglicht das Hinweissignal eine Unabhängigkeit zwischen Entladeüberwachung und der Anwendung beziehungsweise dem Energiemanagement. Die Auflösung und Genauigkeit von Hardware-Lösungen wird zwar nicht erreicht, allerdings wurde nachgewiesen, dass sich der Ansatz zur Steuerung des Energieverbrauchs eignet. Eine Erkennung von Anomalien im Verhalten von Batterien, die sich in der Spannung äußern, ermöglicht ein Gegensteuern. Sind die Anforderungen der Anwendung flexibel genug, den Verbrauch ausreichend zu senken, kann das Lebenszeitziel erreicht werden. Andernfalls wird zumindest die verbleibende Laufzeit verlängert, um eine externe Reaktion zu ermöglichen.

Die Wirkungsweise und Eignung des Ansatzes wurde in verschiedenen Szenarien evaluiert und zeigte insbesondere die Wirksamkeit unter realen Umgebungsbedingungen (Tag-Nacht-Zyklus) mit zwei verschiedenen Batterietechnologien. Temperaturbedingte Spannungsabfälle konnten erkannt werden, was eine Reaktion ermöglichte, die Lebenszeitziele erreichen half. Das vorgestellte Verfahren eignet sich als Basis für energiebewusste Anwendungen oder für ein dynamisches Energiemanagement, da es genutzt werden kann, um die angenommene Energiemenge an das Verhalten der Energieversorgung anzupassen.



# Verbrauchserfassung im Feld

```

graph TD
    EM[Energiemodell] --> VE[Verbrauchserfassung]
    VE --> G[Gerät]
    G --> VE
    VE --> A[Anwendung]
    A --> G
    A --> B[Batterie]
    B --> SS[Spannungssensor]
    SS --> BM[Batterie-manager]
    BM --> A
    BM --> VE
    K[Konfiguration] --> BM
    
```

Während der Ausbringungsphase kann eine Verbrauchserfassung den Verbrauch der Anwendung überwachen und so eine Steuerung vereinfachen. Abbildung 4.1 zeigt die Erweiterung des in Kapitel 3 vorgestellten Ansatzes (vergleiche Abbildung 3.1 auf Seite 51). Die Anwendung ist in der Lage, einzelne Parameter individuell aufgrund des Verbrauchs einzelner Geräte anzupassen. Dieses einfache Energiebewusstsein macht eine Observation von Änderungen des Verbrauchs möglich und gestattet es, den Verbrauch auf der Basis verbleibender Energie zu steuern. Initiale Messungen wie bei einer reinen Batterieüberwachung sind zur Einstellung des Verbrauchs nicht mehr notwendig.

In diesem Kapitel wird ein Ansatz für eine Verbrauchserfassung zur Laufzeit vorgestellt. Dieser ist rein Software-basiert, bezieht jedoch im Gegensatz zu anderen Ansätzen die Spannung und die Effizienz eventuell vorhandener Spannungsregler in die Erfassung ein. Die Granularität der Erfassung lässt sich je nach Anforderung der Anwendung anpassen. Das zugrundeliegende Energiemodell beruht auf Verbrauchswerten, die statistisch aus mehreren gemessenen Systemen gewonnen wurden.

## 4.1 Problembeschreibung

Zu jedem Zeitpunkt  $t$  während der Laufzeit  $T$  eines tief eingebetteten Systems soll es möglich sein, die bis dahin verbrauchte Energie  $\bar{E}(t)$  zu approximieren. Wie in Kapitel 2 dargelegt, hängt der Verbrauch eines Systems von dessen aktivierten Komponenten ab und wird wie in Formel 4.1 dargestellt aus der Summe der Einzelverbräuche der Geräte errechnet.

$$\bar{E}(t) = \sum^{Geräte} \bar{E}_{Gerät}(t) \quad (4.1)$$

Geräte können verschiedene Betriebsmodi besitzen, welche unterschiedliche Lasten verursachen. Somit hängt die verbrauchte Energie eines Gerätes davon ab, wie lange es sich in welchem Betriebsmodus befunden hat (siehe Formel 4.2).

$$\bar{E}_{Gerät}(t) = \sum^{Modi} \Delta t_{Modus} * I_{Modus} * V \quad (4.2)$$

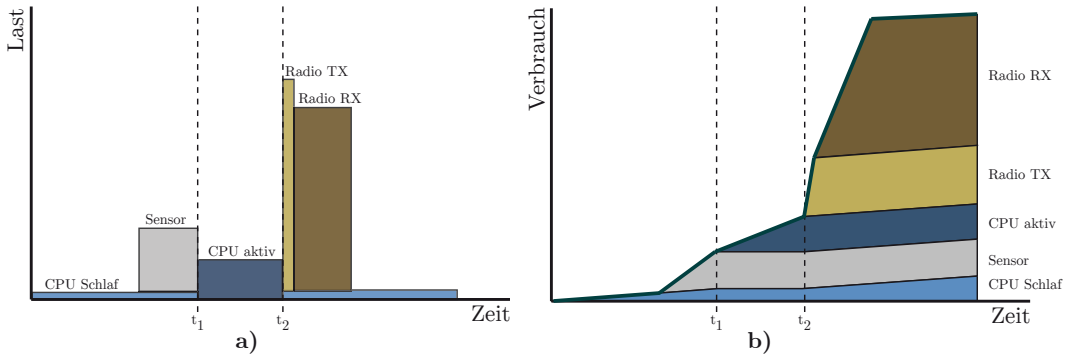


Abbildung 4.2: a) Last durch Geräteaktivitäten und b) daraus folgender Gesamtverbrauch, über die Zeit aufgeteilt in Komponenten

In Abbildung 4.2 sind das typische Geräteverhalten, die daraus resultierende Last sowie der Energieverbrauch eines drahtlosen eingebetteten Sensorsystems beispielhaft dargestellt. Mit jeder Geräteaktivität steigt neben der durch das Gerät verursachten Last der Energieverbrauch, wobei mehrere Geräte auch zur selben Zeit aktiv sein können. Der Verbrauch der aktiven CPU wird, wie in der Abbildung gezeigt,

von der Last  $I_{CPU_{aktiv}}$  und der Dauer  $\Delta t_{CPU_{aktiv}} = t_2 - t_1$  bestimmt. Daraus lässt sich die verbrauchte Energie wie in Formel 4.3 in Abhängigkeit von der Spannung errechnen.

$$\bar{E}_{CPU_{aktiv}}(t) = \Delta t_{CPU_{aktiv}} * I_{CPU_{aktiv}} * V \quad (4.3)$$

Die CPU besitzt jedoch mehrere Modi, in diesem Fall neben dem aktiven einen Schlafmodus. Beide sind gegensätzlich, sodass die CPU entweder aktiv ist oder schläft. Der Verbrauch der CPU setzt sich aus beiden Modi zusammen (Formel 4.4).

$$\bar{E}_{CPU}(t) = \Delta t_{CPU_{aktiv}} * I_{CPU_{aktiv}} * V + (t - \Delta t_{CPU_{aktiv}}) * I_{CPUSchlaf} * V \quad (4.4)$$

Dies gilt analog für alle anderen Komponenten des Systems. Der theoretische Verbrauch der Komponenten ist zwar durch das Datenblatt des Herstellers gegeben, allerdings existieren wie bereits dargelegt Variationen im tatsächlichen Verbrauch. Zusätzlich hat die Energieversorgung einen Einfluss auf den Verbrauch. Wird ein Spannungsregler genutzt, hängt der tatsächliche Verbrauch des Systems von dessen Effizienz ab, welche wiederum von der Versorgungsspannung und der vom System erzeugten Last abhängt. Auch ohne Regler kann der Verbrauch von Komponenten von der Spannung abhängen und sich mit sinkender Versorgungsspannung verringern.

Eine Verbrauchserfassung muss solche Abweichungen mit in Betracht ziehen. Dies ist insbesondere für Ansätze problematisch, die ohne tatsächliche Messung zur Laufzeit auskommen müssen und den Verbrauch rein Software-basiert durch Observation der Geräteaktivität und ein im Vorhinein ermitteltes Energiemodell abschätzen. Ein denkbarer Weg, um in solchen Fällen mit Gerätevariationen umzugehen, ist die Erzeugung eines individuellen Modells für jedes System durch Messung. Das ist jedoch teuer und fehleranfällig. Eine andere Möglichkeit besteht darin, Maximalwerte anzunehmen und für alle Systeme zu nutzen. Zwar ist dies einfacher, aber ungenau, da Ausreißer die Annahme dominieren würden. Für die Mehrzahl der Systeme würde der tatsächliche Verbrauch deutlich kleiner sein als angenommen. Ein weiterer Weg ist es, einen Teil der Systeme zu messen und ein statistisches Modell zu nutzen, um die meisten, aber nicht alle, Systeme abzudecken. Angesichts der möglichen Variationen ist es notwendig, eine dynamische Verbrauchserfassung zu schaffen, die zugrundeliegende Verbrauchsdaten abhängig von der Spannung oder der Effizienz eines Spannungsreglers anpassen kann.

## 4.2 Stand der Technik

Vor der Ausbringung lässt sich eine Abschätzung des Verbrauchs durch Simulation durchführen. Dazu existieren Simulatoren [138, 176, 177], die auf verschiedenen

Ebenen ansetzen und dementsprechend unterschiedliche Granularitäten und Genauigkeiten bieten. Auf diese Weise ist es möglich, eine Vielzahl von Systemen zu simulieren und das Verhalten von Sensornetzen zu evaluieren. Eine gängige Praxis ist es, existierende Simulationsumgebungen mit einem vorher durch Messungen ermittelten Energie- und Zeitmodell zu versehen. Die Zustände der Hardware werden simuliert und mit dem Modell verknüpft. Problematisch ist dabei die Abschätzung des Verbrauchs der CPU, da die Simulation keine Informationen über die Dauer der Ausführung der Applikation und auftretende Unterbrechungen auf der Zielplattform besitzt. In [176] wird dies durch Annotation gelöst, deren Werte durch statische Codeanalyse ermittelt wurden. In [178] wird durch Emulation der tatsächliche Code ausgeführt. Letzteres ist zwar genauer, aber deutlich aufwendiger. Die Einbeziehung von Umweltfaktoren, die einen Einfluss auf den Verbrauch haben, ist in solchen Simulatoren meist nicht vorgesehen.

Zwar lässt sich der Verbrauch näherungsweise durch Simulation vorhersagen, die Durchführung von Messungen auf realer Hardware ist jedoch flexibler und präziser. Wenn der Entwickler keinen Zugriff auf entsprechendes Messequipment besitzt, ist ein Software-basierter Ansatz auf realer Hardware dennoch durchführbar.

Für eine Erfassung des Energieverbrauchs im Feld muss jedes System mit einem geeigneten Mechanismus ausgestattet werden. Dieser kann sowohl Hardware-basiert (Kapitel 4.2.1) als auch rein Software-basiert (Kapitel 4.2.2) oder eine Mischung aus beidem sein (Kapitel 4.2.3). Während eine Hardware-Lösung den Stromverbrauch messtechnisch erfasst, ist eine Software-Lösung darauf angewiesen, die Aktivität des Systems zu observieren und mit geeigneten Verbrauchswerten zu verrechnen.

#### **4.2.1 Hardware-basierte Ansätze**

Es existieren verschiedene Hardware-basierte Ansätze. Allen gemein ist der Mehraufwand hinsichtlich Materialkosten und Energieverbrauch. Sie bieten eine Erfassung des Verbrauchs des Gesamtsystems, womit sie implizit Varianzen im Energieverbrauch erfassen, die durch Spannung, Temperatur oder Fertigung auftreten. Da jedoch nur der Gesamtverbrauch erfasst wird, ist eine Aufschlüsselung auf die einzelnen Komponenten nicht ohne weiteres möglich. Prinzipiell wäre für jede Komponente separate Hardware nötig. Eine Aufspaltung unter gewissen Vorraussetzungen - der Erfassung der Gerätezustände - mithilfe mathematischer Analysemethoden ist jedoch möglich.

##### **Smart Battery Systeme**

Eine einfache Möglichkeit, den Verbrauch während der Laufzeit zu erfassen, bieten die in Kapitel 3.2.1 vorgestellten Smart Battery Systeme. Allerdings sind diese für tief eingebettete Systeme wie Sensorknoten kaum geeignet, wie in [179], [180] und

[181] dargelegt. Dies wird mit der geringen dynamischen Bandbreite, der geringen Geschwindigkeit und dem Energieverbrauch begründet. So sind drahtlose tief eingebettete Systeme die meiste Zeit inaktiv und benötigen Energie nur im Bereich von  $\mu\text{A}$ . Unterbrochen wird dies von Phasen der Aktivität mit Verbräuchen bis zu dutzenden  $\text{mA}$  und einer Dauer von wenigen  $\mu\text{s}$ .

### **Sensor Node Management Device**

In [182] und [181] werden sogenannte Sensor Node Management Devices (SNMD) vorgestellt. Diese sind den Smart Battery Systemen ähnlich und nutzen dieselbe Technik. Der Spannungsabfall über einen Messwiderstand wird durch einen Analog-Digital-Wandler erfasst, um den Strom zu messen und mithilfe der Spannungen und der Zeit den Energieverbrauch zu bestimmen. Im Gegensatz zu den Smart Battery Systemen wurden sie für Sensorknoten entworfen und setzen Komponenten ein, die hinsichtlich der Bandbreite und der Geschwindigkeit geeigneter sind, also AD-Wandler mit hoher Auflösung und Erfassungsrate. Jedoch zielen SNMD nicht auf den Einsatz im Feld, sondern sind für Testaufbauten ausgelegt und verfügen über entsprechende zusätzliche Funktionalität, wie beispielsweise das Umschalten zwischen verschiedenen Energiequellen und Lademanagement von Akkumulatoren.

### **Scalable Power Observation Tool**

Das in [179] vorgestellte Scalable Power Observation Tool (SPOT) beruht zwar wie die SNMD auf einem Messwiderstand, nutzt allerdings einen Frequenz-Spannungswandler in Kombination mit Zählern für Energie und Zeit. Die Unzulänglichkeiten der Analog-Digital-Wandler werden so umgangen. Das ermöglicht eine Erfassung des Energieverbrauchs mit relativ hohen Frequenzen und hoher Auflösung. Die gleichzeitige Erfassung und das Auslesen von Energie und Zeit erlauben eine einfache Erfassung des Verbrauchs innerhalb eines Zeitfensters. Nachteilig an diesem System sind neben den Fertigungskosten die Notwendigkeit einer Kalibrierung und der laut [180] hohe Energieverbrauch von über einem  $\text{mA}$ .

### **iCount**

iCount [180] macht sich die in vielen Systemen bereits vorhandenen Spannungsregler zunutze. Bei vielen Reglern existiert ein in weiten Teilen linearer Zusammenhang zwischen Last und Schalzhäufigkeit, womit bei jedem Schaltvorgang eine identische Energiemenge fließt. Durch Einfügen einer einzigen Leitung werden die Schaltvorgänge durch den  $\mu\text{C}$  observierbar und können durch Nutzung der vorhandenen Zähler mit nur sehr geringem Mehraufwand erfasst werden. Zusätzlich ermöglichen diese

Zähler durch Unterbrechungen bei Wertegleichheit die Implementierung von Kontingenten für energiebasiertes Scheduling.

Nachteilig an diesem Ansatz ist insbesondere der aufwendige Kalibrierungsprozess, da Werte für unterschiedliche Verbräuche gemessen und der Verbrauch durch linearere Regressionsanalyse errechnet werden muss. Da die für Regler notwendigen Kapazitäten eine hohe Fertigungsstreuung besitzen, ist eine Kalibrierung jedes Systems notwendig. Hinzu kommen die in Kapitel 2.1.4 dargelegten Probleme, die durch Nutzung von Spannungsreglern entstehen.

#### **4.2.2 Software-basierte Ansätze**

Ansätze, die ohne Hardware auskommen, können naturgemäß nur eine Abschätzung des Energieverbrauches liefern. Sie sind den reinen Simulationen überlegen, da sie das tatsächliche Verhalten des Systems im Feld observieren. Um das Verhalten erfassen zu können, muss das System durch Einfügen von Observationspunkten an den Stellen modifiziert werden, die den Energieverbrauch verändern. Dies kann auf Grundlage ganzer Anwendungsblöcke geschehen, deren Verbrauch vorher durch Messung ermittelt wurde, oder durch die zeitliche Erfassung der Gerätezustände.

##### **Anwendungsblöcke**

Erfolgt die Verbrauchserfassung auf der Ebene von Anwendungsblöcken, muss der Verbrauch für jeden Block im Voraus ermittelt werden, was zwangsläufig durch den Anwendungsentwickler durchgeführt werden muss. Dies kann durch Messung wie in [169] oder durch Simulation wie in [183] erfolgen. Selbst kleine Veränderungen der Anwendung oder Hardware machen hierbei jedoch aufwendige Anpassungen notwendig.

##### **Zustandserfassung**

Werden die Zustände der Hardware durch Observationspunkte erfasst, kann der Verbrauch mithilfe eines Energiemodells durch die Erfassung der Zeit berechnet werden, die ein Gerät in einem Zustand verbringt. Die Vorgehensweise ähnelt dem der Simulation, sodass ein einmal erzeugtes Modell sowohl für die Simulation, als auch für die Verbrauchserfassung im Feld genutzt werden kann [184].

Da das Energiemodell von der Anwendung unabhängig ist, ist der Aufwand für den Anwendungsentwickler entsprechend gering. Anpassungen erfolgen nur im Betriebssystem und in den Treibern, sodass Änderungen in der Hardware nur geringen Aufwand erzeugen.

Ansätze dieser Art wurden für verschiedene eingebettete Betriebssysteme wie TinyOS [170] oder Contiki [171] entwickelt und ähneln sich in ihrem Aufbau. Neben der



Erfassung des Verbrauchs der Hardwarekomponenten zielt [170] auf eine Erfassung und Abrechnung von Abfragen ab (als Beispiel wird TinyDB [19] in einem Mehrbenutzerszenario genutzt), die über mehrere Soft- und Hardware-Teile und Knoten reichen. Analog dazu zielt [185] auf die Erfassung und Zuweisung von Netzwerkprotokollaktivitäten.

Der Gesamtverbrauch errechnet sich aus der Summe aller individuellen Verbräuche der Geräte und ihrer Zustände. Die Verbrauchsinformationen stehen dem System während der Laufzeit zur Verfügung und können durch ein Energiemanagement beispielsweise zur Anpassung der Anwendung genutzt werden. Da der Verbrauch für jeden Zustand individuell erfasst werden kann, ist eine Anpassung der Anwendung feingranularer möglich, als es mit Hardware-basierten Ansätzen realisierbar ist.

Die Genauigkeit der Ansätze hängt von der Genauigkeit des Energiemodells, von dessen Detailgrad und der zeitlichen Auflösung der Verbrauchserfassung ab. In [186] wurde die Genauigkeit von sogenannten Drei-Zustands-Modellen (Empfangen, Senden und Schlaf) mit verschiedenen Kommunikationsprotokollen untersucht und verbessert. Der tatsächliche Verbrauch wurde mithilfe eines SNMD erfasst, wobei Variationen zwischen Knoten im Bereich von 4,2% beobachtet wurden. Die Autoren verbesserten die Genauigkeit durch mehrere Anpassungen des Energiemodells. In einem ersten Schritt wurde das Modell durch Messung des Gesamtverbrauches und einer Regressionsanalyse erzeugt. Als wesentliche Quelle von Ungenauigkeiten wurde anschließend das Fehlen von Zwischenzuständen ausgemacht, welche daraufhin wiederum durch Regressionsanalyse ermittelt und dem Modell hinzugefügt wurden. Die größte Genauigkeit wurde durch Kalibrierung des Modells für jeden Knoten und jedes Protokoll erreicht.

### 4.2.3 Hybride Ansätze

Hybride Ansätze verbinden den gemessenen Verbrauch mit Observationspunkten in der Software. Das zugrundeliegende Energiemodell wird durch die Messungen erzeugt oder verbessert.

#### Sesame

Um die Unzulänglichkeiten der Smart Battery Systeme auszugleichen, wird in [181] Sesame vorgestellt, ein System zur automatischen Erzeugung von Energiemodellen zur Laufzeit. Dabei werden für Linux-basierte Systeme wie Laptops und Smartphones die per ACPI bereitgestellten Informationen über die Geräte genutzt, um durch eine lineare Regressionsanalyse ein Energiemodell zu erstellen. Der Ansatz ermöglicht eine höhere zeitliche Auflösung bei gleichzeitig größerer Genauigkeit, ist aufgrund der Voraussetzungen jedoch nicht für leistungsschwache tief eingebettete Systeme geeignet.

### Quanto

Quanto [187] ist eine für TinyOS entwickelte Verbrauchserfassung. Für die Erfassung wird iCount [180] genutzt. Es wird ein Labeling vorgestellt, das Aktivitäten erfassen kann, die verschiedene Hardwareaktionen und Knoten betreffen. So ist es möglich, Anfragen innerhalb eines Sensornetzes zu verfolgen. Bei jeder Änderung eines Gerätezustandes und damit potenziell des Energieverbrauchs werden die Daten zu Verbrauch, Systemzustand und Zeit erfasst und gespeichert. Diese Informationen werden nach der Laufzeit genutzt, um durch lineare Regressionsanalyse die verbrauchte Energie den Teilen des Systems zuzuordnen. In der vorliegenden Form steht dem System also zur Laufzeit keine Information über den Verbrauch der einzelnen Teile zur Verfügung, sondern nur der gemessene Verbrauch durch die Hardware. Detaillierte Informationen sind nur im Anschluss verfügbar. Zusätzlich limitiert der zur Verfügung stehende Speicher die Laufzeit des Systems.

### Joule Watcher

Der in [188] präsentierte Ansatz nutzt die in leistungsfähigeren Prozessoren integrierten Performanzzähler, um Rückschlüsse auf die von Prozessen genutzten Hardwareteile zu ziehen. Wie bei den Software-basierten Ansätzen muss ein Energiemodell im Vorhinein erzeugt werden, um mithilfe der Zähler den Verbrauch errechnen zu können. Da diese nur im Prozessor verfügbar sind, muss für andere Systemteile ein anderes Verfahren genutzt werden. Jedoch sinkt der für die Verbrauchserfassung notwendige Mehraufwand durch die Hardwareunterstützung.

#### 4.2.4 Zusammenfassung der Ansätze

Wie Tabelle 4.1 zeigt, haben alle Ansätze zur Ermittlung des Verbrauchs Vor- und Nachteile. Die Bewertung erfolgt dabei nach einem mehrteiligen Maßstab: sehr gut (++), gut (+), neutral (o), schlecht (-) und sehr schlecht (--).

Hardware-basierte Ansätze sind trotz der impliziten Erfassung von Varianzen durch ihre Kosten und den zusätzlichen Energieaufwand für die Verbrauchserfassung im Feld eher ungeeignet. Eine Aufschlüsselung auf einzelne Gerätezustände und Programmteile wie bei hybriden Ansätzen ist nur durch hohen Rechenaufwand möglich, zum Beispiel durch Regressionsanalyse. Dadurch sind sie nur eingeschränkt für das Energiemanagement tief eingebetteter Systeme geeignet.

Software-basierte Ansätze liefern nur eine Abschätzung des Verbrauchs und benötigen ein Energiemodell. Gerade dies ist für die Genauigkeit der Abschätzung essentiell, da die Granularität der betrachteten Hardwarezustände und die Qualität der Verbrauchswerte die Genauigkeit bestimmen. Modelle bisheriger Ansätze basieren meistens auf der Messung nur weniger Knoten. Die größte Genauigkeit wird bei

Tabelle 4.1: Bewertung der verschiedenen Ansätze zur Verbrauchsermittlung im Feld

	Smart Battery [151, 152]	SNMD [182, 181]	SPOT [179]	iCount [180]	Anwendungsblöcke [169, 183]	Zustandserfassung [184, 170, 171, 185]	Sesame [181]	Quanto [187]	Joule Watcher [188]
Granularität	--	--	--	--	+	++	+	++	+
Genauigkeit	o	+	++	+	-	-	++	x <sup>2</sup>	-
Kosten	-	--	--	++/-- <sup>1</sup>	++	++	x <sup>2</sup>	x <sup>2</sup>	++
Energieverbrauch	--	--	--	++/-- <sup>1</sup>	++	++	x <sup>2</sup>	x <sup>2</sup>	++
Vorbereitung	++	++	-	+	--	-	++	-	--
Laufzeitaufwand	++	++	++	++	o	o	--	--	++

<sup>1</sup> abhängig ob Spannungswandler bereits im System vorhanden

<sup>2</sup> abhängig von verwendeter Hardwarelösung

einer per-Knoten-Kalibrierung erreicht, was einen nicht vertretbaren Aufwand darstellen kann. Varianzen im Energieverbrauch durch Spannungsschwankungen und die wechselnde Effizienz von Spannungsreglern werden nicht betrachtet.

Da ein Energiemanagement auf detaillierte Informationen über den Verbrauch angewiesen ist, kommen nur Software-basierte Ansätze auf der Basis von Zustandserfassung in Frage. Jedoch muss eine Anpassung hinsichtlich der Variationen erfolgen, um eine akzeptable Genauigkeit für viele Systeme zu erreichen.

### 4.3 Verbrauchserfassung auf Geräteebene

Der im Folgenden vorgestellte Ansatz erweitert existierende Software-basierte Ansätze hinsichtlich Verbrauchsvariationen. Unterschiede, die durch Herstellungsvarianzen entstehen, werden ausgeglichen, indem statistisch erzeugte Verbrauchsdaten genutzt werden, um so ohne die Annahme pessimistischer Werte eine möglichst hohe Abdeckung zu erreichen. Im Gegensatz zu bisherigen Ansätzen ist das Verbrauchsmodell nicht statisch, sondern bezieht dynamisch zur Laufzeit Varianzen ein, die durch Änderungen der Versorgungsspannung auftreten, wie die von der Spannung und Last abhängige Effizienz etwaiger Spannungskonverter.

Zum Zweck der Verbrauchserfassung muss für einen Software-basierten Ansatz existierender Quellcode modifiziert werden. Dies setzt ein Verständnis über die Funk-

tionsweise der Gerätetreiber voraus. Für neuen Treibercode ist dieses Verständnis inhärent gegeben, sodass der Mehraufwand für den Entwickler gering ist.

### 4.3.1 Verbrauchsmodell

Dieser Ansatz basiert wie beispielsweise [171] auf der Erfassung der aktiven Zeit von Gerätezuständen und Modi. Das macht es möglich, den Verbrauch eines Systems auf einfache und flexible Weise zu modellieren und zu erfassen. Um den Energieverbrauch eines Gerätes zu ermitteln, muss neben der aktiven Zeit die durchschnittliche Stromaufnahme bekannt sein. Dazu wurde Quanto's [187] *energy sinks* (Energiesenken) und *power states* (Verbrauchszustände) Sicht adaptiert und an die Bedürfnisse einer Verbrauchserfassung zur Laufzeit angepasst. Für diesen Ansatz ist eine Energiesenke eine potenziell unabhängige (Sub-)Komponente, die Energie verbraucht. Ein Verbrauchszustand bestimmt, wie viel Last von einer Senke erzeugt wird. Eine Senke muss dabei nicht notwendigerweise mit einer Hardwarekomponente übereinstimmen, sondern kann je nach Bedarf auch einzelne Betriebszustände eines Gerätes umfassen.

Die Energiesenken und Verbrauchszustände eines Gerätes bilden einen endlichen Zustandsautomaten, der das energetische Verhalten des Gerätes abbildet. Da in diesem Ansatz die Granularität der verfügbaren Informationen von der Zahl der Senken abhängt, kann es sinnvoll sein, Energiesenken durch Verbrauchszustände zu ersetzen und umgekehrt. Abbildung 4.3 zeigt Beispiele der möglichen Erfassungsgranularität. Für die Entwicklung und den Test ist eine detaillierte Aufschlüsselung der Energieverteilung in der CPU möglicherweise dienlich, um *Energielöcher* zu finden, welche durch falsches Systemverhalten wie einen falsch gewählten Schlafmodus Energie verschwenden (Abbildung 4.3a). Während der produktiven Ausbringung kann eine solch detaillierte Aufschlüsselung für das Energiemanagement oder die Wartung des

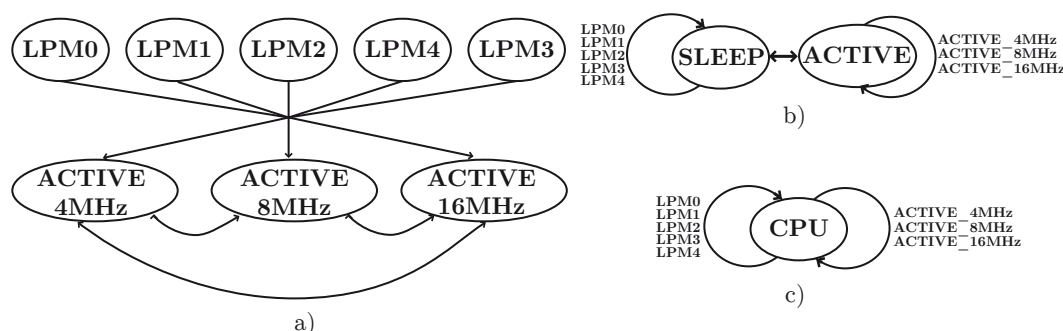


Abbildung 4.3: Mögliche Granularitäten der Erfassung des µC: a) jeder Schlafmodus und die mögliche Frequenzen des aktiven Modus werden individuell in Senken erfasst, b) keine Unterscheidung zwischen den einzelnen Schlafmodi und Frequenzen, c) alle Modi in einer einzelnen Senke erfasst

Systems unnötig sein (Abbildung 4.3b). Wenn der Speicher limitiert ist, kann eine einzelne Energiesenke mit vielen Verbrauchszuständen genutzt werden (Abbildung 4.3c). Dies macht zwar den Gesamtverbrauch des Systems verfügbar, enthält aber keinerlei Informationen über die interne Verteilung. Um eine solche Flexibilität zu ermöglichen, muss der Ansatz fähig sein, zwischen Energiesenken zu wechseln und den Verbrauch einer Senke dynamisch anzupassen.

Änderungen der Versorgungsspannung durch den Spannungsabfall der Energieversorgung können eine adaptive Anpassung des Verbrauchs einer Senke notwendig machen. Zu diesem Zweck überwacht das System die Spannung und informiert involvierte Gerätetreiber, wenn sich die Spannung substanziell geändert hat. Der Treiberentwickler ist für die Bereitstellung der Funktion verantwortlich, die den Verbrauch der Gerätesenken entsprechend anpasst. Die Häufigkeit, in der die Spannung überprüft wird, kann je nach Last angepasst werden, denn hohe Lasten können zu Spannungsabfällen führen und so den Verbrauch der aktiven Geräte verändern. Mit Informationen über die aktiven Geräte kann die Frequenz entsprechend erhöht werden, wenn die Last steigt, und beim Anliegen von wenig Last verringert werden, um Energie zu sparen.

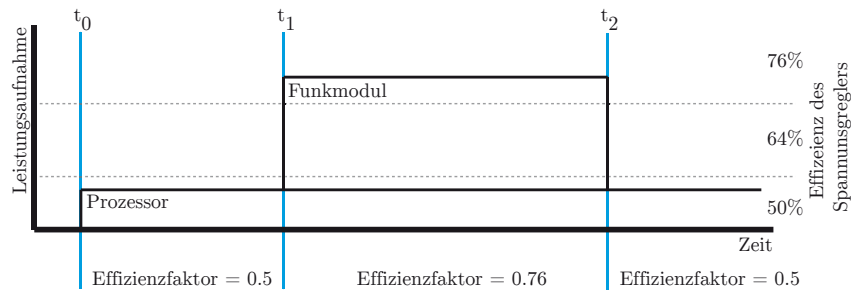


Abbildung 4.4: Beispiel: Einfluss der Last auf die Effizienz eines Spannungsreglers

Die erfasste Energiemenge kann mithilfe eines Effizienzfaktors, der durch einen Spannungsregler verursacht wird, angepasst werden. Wenn ein Regler im System vorhanden ist, ist zwar die Versorgungsspannung stabil, allerdings hängt die Effizienz des Reglers von der Eingangsspannung und der Last ab. Der Effizienzfaktor muss immer dann neu berechnet werden, wenn sich die Eingangsspannung des Spannungsreglers oder die Last ändert. In dem in Abbildung 4.4 dargestellten Beispiel muss der Verbrauch des Prozessors sowohl bei  $t_1$  als auch bei  $t_2$  berechnet werden, da das Funkmodul durch seine Last die Effizienz und damit den Effizienzfaktor verändert. Dieser Effizienzfaktor muss in alle Berechnungen für die Verbrauchserfassung einbezogen werden. In diesem Beispiel berechnet sich der Verbrauch des Prozessors zum Zeitpunkt  $t_2$  wie in Formel 4.5 dargestellt aus zwei Teilen, dem ersten Intervall

von  $t_0$  bis  $t_1$  mit einem Effizienzfaktor von 0,5 und dem zweiten Intervall von  $t_1$  bis  $t_2$  mit einem durch die gleichzeitige Last des Funkmoduls höheren Faktor von 0,76.

$$\bar{E}_{Prozessor}(t_2) = (t_1 - t_0) * I_{Prozessor} * V * \frac{1}{0,5} + (t_2 - t_1) * I_{Prozessor} * V * \frac{1}{0,76} \quad (4.5)$$

Die Berechnung des Verbrauchs des Funkmoduls ist in diesem Beispiel einfacher (siehe Formel 4.6), da dieses nur innerhalb eines Intervalls mit einem Effizienzfaktor aktiv ist.

$$\bar{E}_{Funkmodul}(t_2) = (t_2 - t_1) * I_{Funkmodul} * V * \frac{1}{0,76} \quad (4.6)$$

### 4.3.2 Erzeugung der Verbrauchsdaten

Mithilfe der Verbrauchsmessungen, die in Kapitel 2.3 vorgestellt wurden, können Verbrauchsmodelle konstruiert werden, die als Eingabe für die Verbrauchserfassung dienen. Tabelle 4.2 zeigt diese Modelle. Die aus dem Datenblatt extrahierten Werte wurden für die Konstruktion eines Modells (*m-Datenblatt*) genutzt. Die aus dem Experiment gewonnenen Daten wurden für zwei Modelle verwendet. Ein Modell (*m-Median*) wurde aus dem Median der Experimentdaten gebildet. Zusätzlich wurde ein pessimistisches Modell (*m-Pessimistisch*) erstellt, das auf dem 0,8 Perzentil der Werte beruht. Das heißt, 80% der gemessenen Systeme haben einen geringeren Verbrauch, 20% einen höheren. Auf diese Weise kann das pessimistische Modell sicherstellen, dass der Verbrauch für eine Mehrzahl der Systeme nicht unterschätzt wird, da dies für eine Berechnung der Laufzeit gefährlicher ist als eine Überschätzung. Eine Nutzung der Werte des schlechtesten Systems ist viel zu pessimistisch, da dieses ein Mehrfaches anderer Systeme verbraucht. Das verwendete 0,8 Perzentil überdeckt die meisten Systeme, wobei gute Systeme mit geringem Verbrauch nicht zu stark überschätzt werden.

Wie die Tabelle zeigt, variieren die Unterschiede zwischen den Modellen je nach Betriebsmodus. Die beiden Modelle, die auf den Experimenten beruhen, weichen überwiegend nur geringfügig voneinander und von den Werten des Datenblattes ab. Die Auswirkung dieser Unterschiede hängt vom Nutzungsverhalten der Hardware ab. So hat der Schlafmodus bei Systemen mit einer geringen Anwendungsfrequenz einen substanziellen Anteil am Gesamtverbrauch.

Die Verbrauchserfassung greift aus Effizienzgründen nicht auf Fließkommazahlen zurück und ermöglicht als kleinste erfassbare Einheit ein  $\mu A$ . Da aufgerundet wird, erfolgt eine Überschätzung des Verbrauchs im Schlafmodus. Wie bereits dargestellt, ist eine geringe Überschätzung jedoch weniger gefährlich als eine Unterschätzung des Verbrauchs. Zusätzlich macht es die Verwendung von Integerzahlen möglich, den Be-

Tabelle 4.2: Verbrauchsmodelle auf Grundlage der ermittelten Stromaufnahme

	Durchschnittliche Stromaufnahme in mA		
	m-Datenblatt	m-Median	m-Pessimistisch
LPM3 Schlaf	0,0022	0,0028 (+27,3%)	0,0044 (+100%)
CPU@8Mhz	1,75	1,755 (+0,3%)	1,770 (+1,2%)
CPU@16Mhz	3,45	3,369 (-2,4%)	3,407 (-1,2%)
ADC	0,15	0,124 (-17,4%)	0,127 (-15,4%)
REF	0,1	0,105 (+5%)	0,109 (+9%)
Radio IDLE	1,7	1,657 (-2,5%)	1,672 (-1,7%)
Radio RX	16,0	16,214 (+1,3%)	16,371 (+2,3%)
Radio TX@0dBm	17,8	17,694 (-0,6%)	17,894 (+0,5%)
Radio TX@10dBm	33,0	34,441(+4,4%)	35,952 (+8,9%)

rechnungsaufwand durch Nutzung von oftmals vorhandenen Hardwaremultiplicierern zu senken.

## 4.4 Evaluation

Für die Evaluation wurde der Ansatz zur Verbrauchserfassung für das Betriebssystem REFLEX auf der MSP430 Plattform implementiert. Im Folgenden wird zunächst der durch die Verbrauchserfassung notwendige Mehraufwand betrachtet. Anschließend wird die Genauigkeit des Ansatzes und der Modelle unter Nutzung des in Kapitel 2.3.1 vorgestellten Messaufbaus untersucht. Daran schließen sich Szenarien an, welche die Funktion der dynamischen Systembestandteile überprüfen und die Verwendung der Verbrauchserfassung demonstrieren.

### 4.4.1 Mehraufwand

Da im Zuge der Verbrauchserfassung die Zustandsänderungen der Hardware beobachtet werden, entsteht ein Mehraufwand sowohl hinsichtlich des benötigten Speichers als auch hinsichtlich der Programmausführung.

Der Mehrbedarf an Speicher wird in Tabelle 4.3 für die Komponenten der Verbrauchserfassung aufgeschlüsselt. Der für jedes erfasste Gerät notwendige Speicher im RAM wird von der Zahl der Senken bestimmt. Der Speicherbedarf im ROM beträgt mit ca. 3kB weniger als 10% des zur Verfügung stehenden Speichers der verwendeten Plattform. Dieser Mehrbedarf ist zwar signifikant, liegt jedoch in dem Bereich, der von Kommunikationsprotokollen oder umfangreichen Treibern benötigt wird. Durch die Aufteilung in Einzelkomponenten ist es außerdem möglich, den

Tabelle 4.3: Speicherbedarf für einzelne Komponenten der Verbrauchserfassung für die MSP430 Plattform und den gcc Compiler: Listengröße  $n$ ,  $x$  repräsentiert die Anzahl der Senken.

	Speicherbedarf in Byte	
	ROM	RAM
energyAccountingManager <sup>1</sup>	800	$16+n*2$
consumerBase Funktionen	1042	0
consumerBase pro Gerät	$40+\text{Funktionsaufrufe}$	$20+x*10$
supplyVoltageObserver	372	$52+n*2$
converterEfficiencyObserver <sup>2</sup>	758	58

<sup>1</sup> inklusive Treiber für HW-Multiplikation

<sup>2</sup> mit sechs Effizienzstufen

Speicherbedarf individuell zu verringern, wenn keine Abhängigkeit von einem Spannungsregler besteht oder der Einfluss der Versorgungsspannung vernachlässigbar ist.

Der Laufzeitaufwand der Verbrauchserfassung wird hauptsächlich durch die Zeiterfassung, -differenzbildung und die Multiplikation mit den Verbrauchswerten verursacht. Wesentliche Einflussfaktoren für den Mehraufwand sind die Anzahl an Geräteaktivitäten und die Zeitpunkte, zu denen im Programmablauf die Methoden der Verbrauchserfassung aufgerufen werden. Der Zustandswechsel der Hardware kann nach der Verbrauchserfassung erfolgen (und ist somit pessimistisch) oder vorher (optimistisch). Da nach einem Zustandswechsel Totzeiten existieren können, weil beispielsweise auf das Ende der Hardware-Aktivität gewartet wird, hat bei der optimistischen Verbrauchserfassung der Mehraufwand keinen Einfluss auf das zeitliche Verhalten.

Um den Mehraufwand der Verbrauchserfassung einschätzen zu können, wurde eine Anwendung genutzt, die das Ping-Pong-Verfahren umsetzt. Es werden kontinuierlich Pakete versendet und anschließend auf Antwort gewartet, um dann erneut zu senden. Jeder Zyklus besteht aus mehreren Zustandswechseln, jeweils drei des Funkmoduls und des Prozessors. Zwei FeuerWhere Sensorknoten [189] wurden verwendet, um die Zahl der Pakete pro Sekunde mit und ohne Verbrauchserfassung bei unterschiedlichen Frequenzen des Prozessors zu ermitteln.

Tabelle B.2 im Anhang zeigt die vollständigen Ergebnisse des Experimentes. Die prozentuale Verringerung der Pakete pro Sekunde und somit der Mehraufwand ist in Abbildung 4.5 sowohl bei der Verwendung der optimistischen als auch der pessimistischen Verbrauchserfassung dargestellt. Durch die geringere Paketrate bei größeren Paketen werden weniger Zustandswechsel durchgeführt. Die Zahl der Zustandswechsel beeinflusst den Mehraufwand. Der Zeitpunkt der Verbrauchserfassung spielt für den Mehraufwand wie angedeutet ebenso eine Rolle. Die optimistische Verbrauchs-



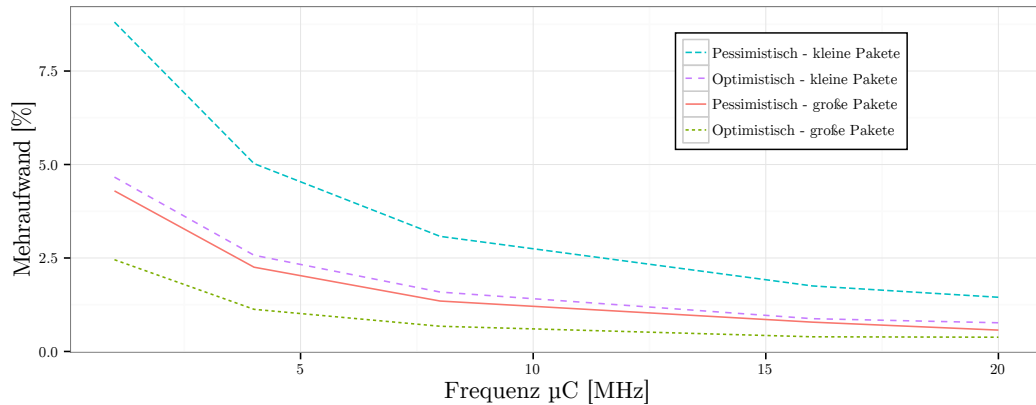


Abbildung 4.5: Mehraufwand der Verbrauchserfassung bei Funkanwendung bei zwei Paketgrößen und verschiedenen Frequenzen des Mikrocontrollers: Pessimistische Verbrauchserfassung erfasst vor HW-Zustandswechsel, optimistische danach

serfassung hat einen spürbar geringeren Einfluss auf die Paketrate. Mit steigender Frequenz wird der Einfluss der Verbrauchserfassung geringer. Ab einer Frequenz von 16 MHz ist die Paketrate mit Verbrauchserfassung nur um rund 1% geringer.

Ist im System ein Spannungsregler vorhanden, steigt der Mehraufwand. Bei jedem Zustandswechsel muss überprüft werden, ob sich der aktuelle Effizienzfaktor verändert. Sollte sich die Effizienz ändern, müssen alle aktiven Energiesenken mit dem bisherigen Faktor abgerechnet werden. Das vorangegangene Experiment wurde um die Einbeziehung der Effizienz erweitert. Um den Mehraufwand zu ermitteln, wurde zum einen eine zwangsweise Anpassung des Effizienzfaktors mit daraus folgender Abrechnung aller Senken bei jedem Zustandswechsel durchgeführt und zum anderen die reine Ermittlung des Faktors ohne eine zusätzliche Abrechnung weiterer Senken.

Tabelle B.3 im Anhang zeigt die gesamten Ergebnisse. Der Mehraufwand der Verbrauchserfassung unter Einbeziehung der Effizienz ist in Abbildung 4.6 als prozentuale Verringerung der Pakete pro Sekunde dargestellt. Der Einfluss der Frequenz und Paketgröße ist dabei ähnlich wie bei der reinen Verbrauchserfassung (siehe Abbildung 4.5). Während die reine Berechnung der Effizienz den Mehraufwand bei den höheren Arbeitsfrequenzen nur geringfügig erhöht, steigt dieser bei der vollständigen Abrechnung deutlich, liegt aber in jedem Fall unter 3%. Bei geringeren Arbeitsfrequenzen ist der Mehraufwand nahezu doppelt so hoch wie bei der reinen Verbrauchserfassung und liegt im Extremfall bei über 15%. Ein so hoher Mehraufwand wirkt sich potenziell auf den Energieverbrauch aus. Durch die Berechnungen werden Geräte möglicherweise verspätet abgeschaltet und sind somit länger aktiv.

Das verwendete Szenario stellt einen Extremfall dar. In einer realen Anwendung, die hohe Laufzeiten erreichen soll, finden nur wenige Aktivitäten unterbrochen von

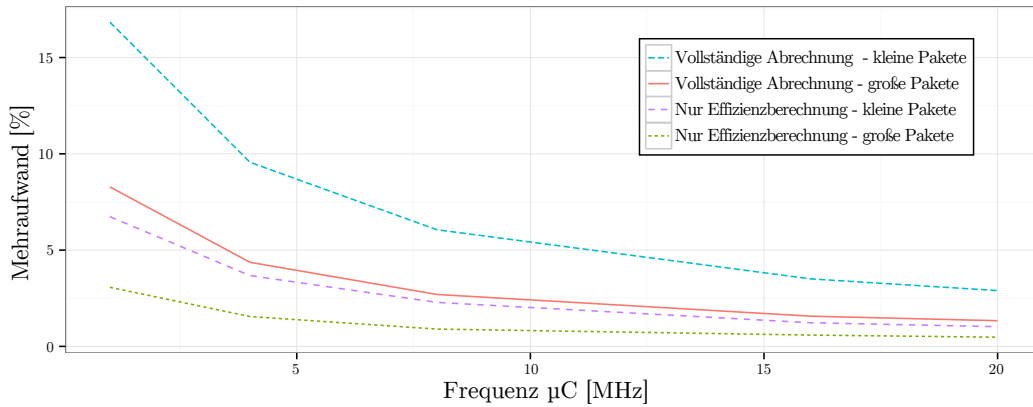


Abbildung 4.6: Mehraufwand der Verbrauchserfassung bei Anpassung an die Effizienz eines Spannungskonverters: Bei vollständiger Abrechnung werden bei Zustandswechsel alle aktiven Senken abgerechnet. „Nur Effizienzberechnung“ verzichtet auf Anpassung der Effizienz.

langen Phasen der Inaktivität statt, sodass der Mehraufwand durch die geringe Zahl an Zustandsübergängen vernachlässigt werden kann. Zusätzlich muss nicht zwangsläufig bei jedem Zustandswechsel die Effizienz des Konverters angepasst werden, was den Mehraufwand in einer realen Anwendung weiter reduziert. Der Mehraufwand lässt sich durch die Wahl der Position für die Funktionen der Verbrauchserfassung verringern. Grundsätzlich lässt sich die Verbrauchserfassung weiter optimieren, beispielsweise indem die Berechnung des Verbrauchs nur dann durchgeführt wird, wenn die Werte benötigt werden. Dies ist insbesondere dann sinnvoll, wenn keine Hardware-Multiplikation zur Verfügung steht.

#### 4.4.2 Vergleich von Modell und Messung

Ein Energiemanagement oder eine energiebewusste Anwendung trifft ihre Entscheidungen auf Grundlage der zur Verfügung stehenden Informationen über den Verbrauch und die zur Verfügung stehende Restenergie. Die Verlässlichkeit dieser Informationen als Abweichung der Verbrauchserfassung und ihrer Modelle von der tatsächlichen Stromaufnahme beeinflusst unmittelbar die Managemententscheidungen. Dabei ist nicht nur der Umfang der Abweichung von Bedeutung. Eine Unterschätzung des Verbrauchs sollte vermieden werden, da dies, wie dargestellt gefährlicher als eine Überschätzung ist.

Zur Abschätzung der Abweichung wurde eine Anwendung implementiert, welche einen Sensor abfragt (die Versorgungsspannung des Systems durch den Analog-Digital-Wandler) und anschließend die Verbrauchsdaten an eine Basisstation überträgt. Die Übertragung erfolgte per Funk und bestand aus einer Sende- und einer Bestätigungsphase. Das Anwendungsintervall, das heißt, der Abstand der Aktivitäts-

ausführung, reichte von 50ms bis 100s, sodass die durchschnittliche Stromaufnahme in einem Bereich von mehreren mA bis wenigen  $\mu\text{A}$  lag. Die vergleichende Messungen erfolgten mit dem in Abbildung 2.16 auf Seite 40 dargestellten Aufbau.

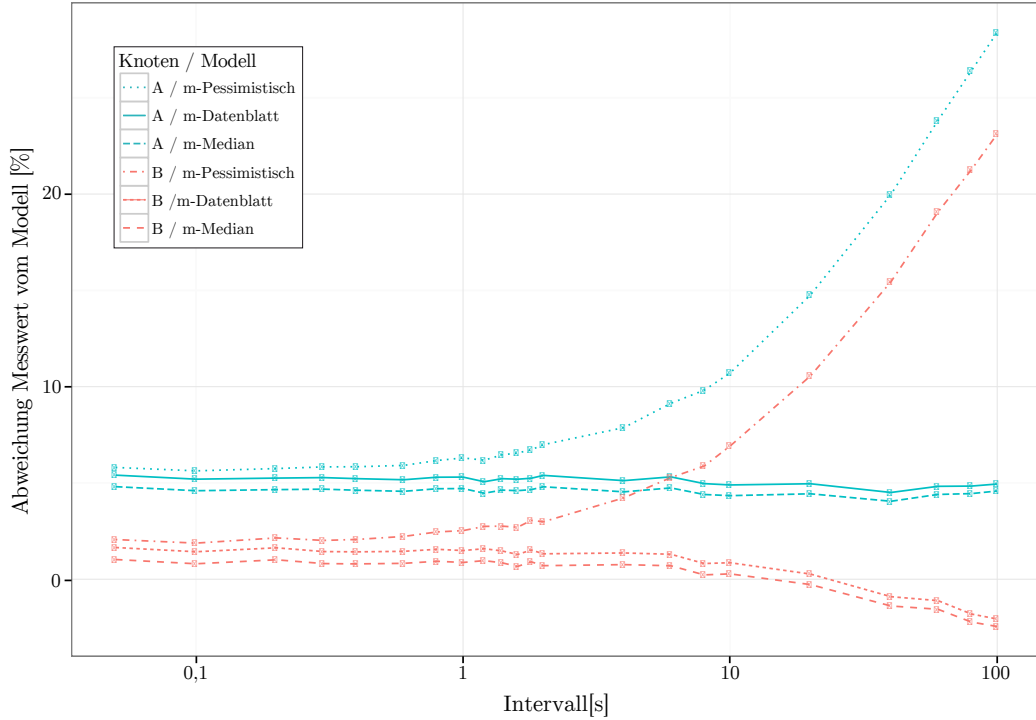


Abbildung 4.7: Prozentuale Abweichung der Modelle von der gemessenen Stromaufnahme bei 3V zweier Knoten (A und B): Das Anwendungsintervall ist in logarithmischer Skala dargestellt.

Abbildung 4.7 zeigt die ermittelte Abweichung der Modelle von den gemessenen Werten zweier Knoten. Knoten A hat dabei für alle Intervalle eine geringere Stromaufnahme als von den Modellen vorhergesagt. Die Abweichung des *m-Median* Modells ist dabei mit  $\approx 4,5\%$  am geringsten und stabil. Ebenso stabil ist das *m-Datenblatt* Modell, mit einer geringfügig höheren Abweichung von  $\approx 5,5\%$ . Mit steigendem Intervall sinkt der Einfluss der aktiven Modi und der Anteil des Schlafmodus nimmt zu. Da dieser für das *m-Pessimistisch* Modell um 66% höher veranschlagt wird, steigt die Abweichung für hier mit zunehmendem Intervall bis auf  $\approx 28\%$ .

Knoten B hat eine größere Stromaufnahme als Knoten A, wodurch die Abweichung bei geringeren Intervallen mit  $\approx 1$  bis 2,5% deutlich geringer ist. Jedoch reichen mit steigendem Einfluss des Schlafmodus sowohl das *m-Datenblatt* als auch das *m-Median* Modell nicht mehr aus, um die Stromaufnahme der Knoten nicht zu unterschätzen. Analog zu Knoten A steigt die Abweichung von Messung und Modell mit dem *m-Pessimistisch* Modell bei zunehmendem Intervall deutlich, ist aber durch

die höhere Stromaufnahme von Knoten B im Schlafmodus mit  $\approx 23\%$  nicht ganz so stark.

Die Ergebnisse zeigen den Einfluss des Schlafmodus und der durch die Implementierung verursachten Überschätzung. Bei niedrigen Anwendungsintervallen und dementsprechend hohem Energieverbrauch ist das *m-Median* Modell in der Lage, den Verbrauch ausreichend genau vorherzusagen. Bei hohen Anwendungsintervallen kann dieses Modell eine Unterschätzung nicht mehr ausschließen, im Falle von Knoten B ist diese mit  $\approx 1,6\%$  gering und lässt sich möglicherweise, wie in Abbildung 4.1 gezeigt, durch einen kombinierten Energiemanagementansatz ausgleichen. Dies gilt ebenso für die Überschätzung des Verbrauchs, wobei auch die durch das pessimistische Modell verursachte deutliche Überschätzung potenziell ausgeglichen werden kann.

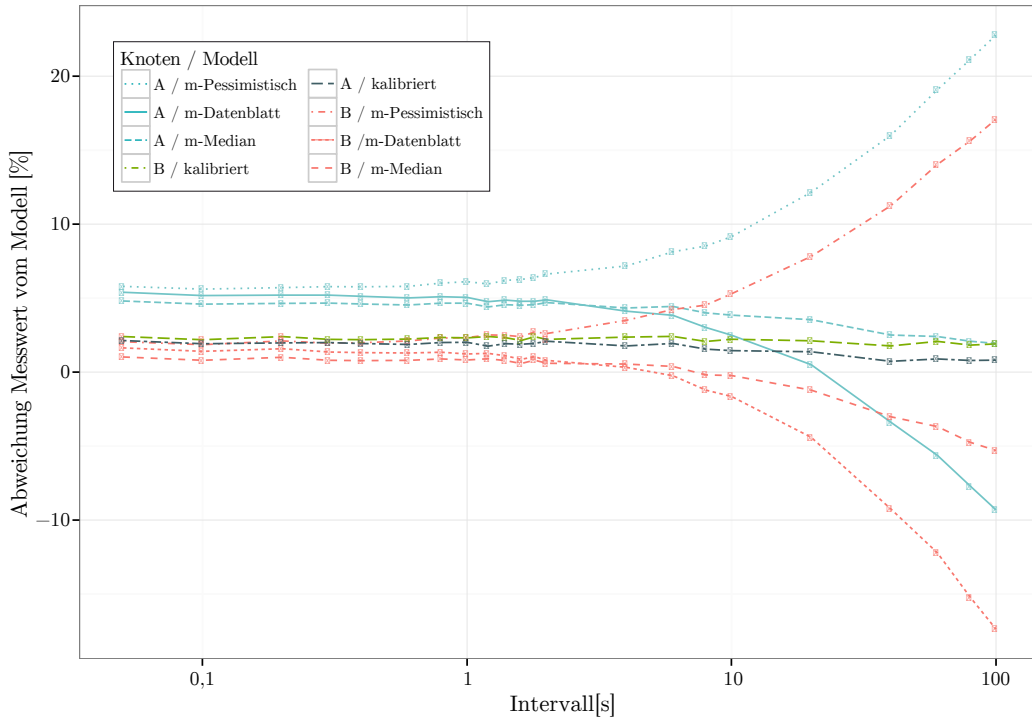


Abbildung 4.8: Prozentuale Abweichung der Modelle mit hoher Auflösung sowie mit per-Knoten-Kalibrierung von der gemessenen Stromaufnahme bei 3V zweier Knoten (A und B): Das Anwendungsintervall ist in logarithmischer Skala dargestellt.

Um die Überschätzung der Modi durch Runden auf den nächst höheren  $\mu\text{A}$  Wert zu verringern, kann die Verbrauchserfassung mit höherer Auflösung angewendet werden. Abbildung 4.8 zeigt die dadurch entstehende Abweichung von den gemessenen Werten der Knoten A und B.

Der Unterschied zwischen den Modellen, die nur eine  $\mu\text{A}$  Auflösung besitzen, und den höher aufgelösten Modellen ist bei kleinerem Anwendungsintervall unwesentlich. Erst mit steigendem Intervall und dadurch größer werdendem Einfluss des Schlafmodus lassen sich Unterschiede erkennen. Das Modell *m-Datenblatt* unterschätzt durch die Abweichung des tatsächlichen Verbrauchs im Schlafmodus von den Datenblattwerten die Stromaufnahme von Knoten A. Zusätzlich nimmt die Abweichung für das *m-Datenblatt* Modell bei diesem Knoten mit steigendem Intervall ab. Für Knoten B nimmt die Abweichung im Vergleich zu den niedriger aufgelösten Modellen zu.

Eine geringere Abweichung, ohne dabei den Verbrauch zu unterschätzen, lässt sich mit Modellen erreichen, die für jeden Knoten separat erzeugt werden. Dies ist zwar aufwendig, liefert aber die besten und stabile Ergebnisse, wie in Abbildung 4.8 dargestellt.

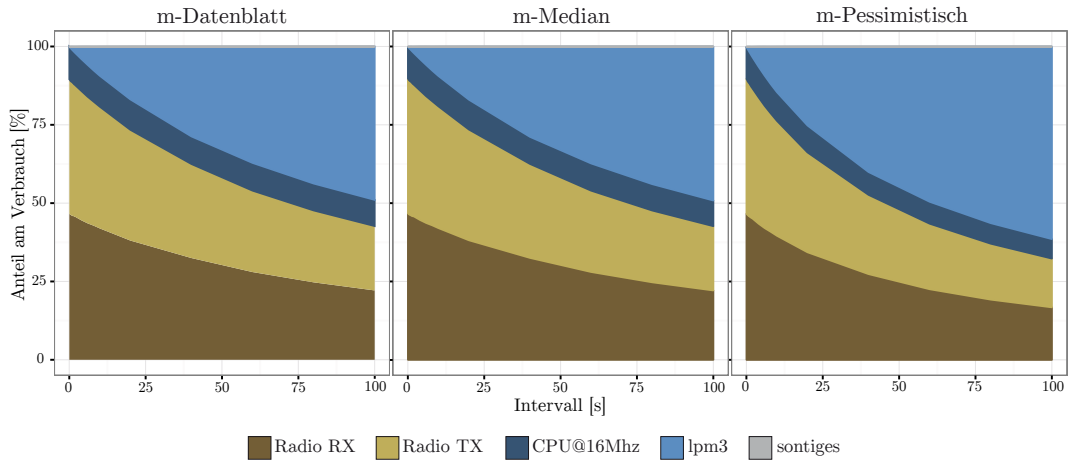


Abbildung 4.9: Beitrag der Geräte/Modi zum gesamten Verbrauch für jedes Modell

In Abbildung 4.9 ist der vorhergesagte Verbrauch für die Modelle aufgeschlüsselt. Wie erwartet, steigt der Einfluss des Schlafmodus mit größer werdendem Anwendungsintervall. Bis auf den Anteil der aktiven CPU, der nahezu gleich bleibt, sinken die Anteile der anderen Modi entsprechend. Da die Unterschiede zwischen den Modellen gering sind, wie in Tabelle 4.2 dargestellt, unterscheiden sich die Anteile der einzelnen Geräte/Modi nur geringfügig. Am deutlichsten weicht die Verteilung beim pessimistischen Modell ab, da dort die Stromaufnahme im Schlafmodus höher veranschlagt wird. Die anderen Modi haben einen entsprechend kleineren Anteil am Verbrauch.

Die Ergebnisse zeigen, dass die Mechanismen der Verbrauchserfassung funktionieren und die verfügbare Granularität Einblicke in das Systemverhalten gestattet. Die Modelle sind abhängig von der durchschnittlichen Stromaufnahme geeignet, den Verbrauch ausreichend genau zu erfassen. Der durch die Verwendung einer Auflösung im Bereich von  $\mu\text{A}$  eingeführte Pessimismus führt zwar zu einer Überschätzung des Verbrauchs, gleicht dadurch aber Varianzen zwischen unterschiedlichen Knoten

aus. Um eine Unterschätzung des Verbrauchs gänzlich auszuschließen, ist eine sorgfältige Komponenten- und Systemauswahl oder aber eine Messung jedes Systems zumindest für den Schlafmodus notwendig.

#### 4.4.3 Adaption an die Versorgungsspannung

Hat ein System Komponenten, deren Verbrauch merklich von der Versorgungsspannung beeinflusst wird, müssen die Verbrauchszustände der Komponenten an die momentane Spannung angepasst werden. Um diese Adaption zu überprüfen, wurde ein System mit einer LED ausgestattet, die alle 500ms an- oder ausgeschaltet wurde. Die Anwendung überprüfte die Versorgungsspannung jede Sekunde. Die Verbrauchsdaten wurden alle 10 Sekunden per Funk übertragen. Da keine statistischen Daten über den Verbrauch der LED verfügbar waren, wurde der Verbrauch der LED gemessen. Das *m-Median* Modell wurde mit der gemessenen durchschnittlichen Stromaufnahme bei 3,0V und 2,2V verglichen. Zusätzlich wurde ein Durchlauf durchgeführt, bei dem die Spannung in Schritten von 0,2V von 3,0V auf 2,2V abgesenkt wurde.

Tabelle 4.4: Gemessene durchschnittliche Stromaufnahme im Vergleich mit den durch die Verbrauchserfassung ermittelten Werten bei verschiedenen Spannungen für System A

	Durchschnittliche Stromaufnahme in mA	
	<i>m-Median</i>	Gemessen
3,0V	1,81	1,8
2,2V	0,75	0,74
3,0V bis 2,2V	1,28	1,27

Tabelle 4.4 zeigt die Ergebnisse der Messungen. Für alle gemessenen Spannungsszenarien beträgt der Unterschied zwischen den adaptiv durch die Verbrauchserfassung ermittelten und dem gemessenen Verbrauch weniger als 1%. Dies ist hauptsächlich durch das Wissen über die durchschnittliche Stromaufnahme der LED zu erklären, da diese einen großen Anteil am Gesamtverbrauch hat. Die Ergebnisse zeigen jedoch, dass die Adaption wie gewünscht arbeitet.

#### 4.4.4 Adaption an Spannungsregler

Wie bereits beschrieben hängt die Effizienz eines Spannungsreglers von der Versorgungsspannung und der vom System erzeugten Last ab. Bei Veränderung eines dieser Parameter (insbesondere einer Lastveränderung durch das An- oder Abschalten von Komponenten) müssen die Verbrauchszustände aller aktiven Komponenten mit dem Effizienzfaktor multipliziert werden. Dieser Faktor wird auf Grundlage der

Spannung und der Last erzeugt. Um dieses Verhalten zu validieren, wurde eine Anwendung genutzt, welche das Funkmodul und den aktiven CPU Modus verwendet und so verschiedene Lasten und somit Effizienzfaktoren erzeugt. Das System verblieb für 500ms im aktiven Modus und wartete anschließend für weitere 100ms auf Nachrichten. Zusätzlich wurde der Systemstatus mit maximaler Sendestärke übertragen.

Als Spannungsregler wurde ein ON Semiconductor NCP1400 [52] genutzt. Die gemessene Effizienz wurde in sechs Schritte ( $<100\mu\text{A}$ ,  $<1\text{mA}$ ,  $<3\text{mA}$ ,  $<10\text{mA}$ ,  $<20\text{mA}$  und  $>20\text{mA}$ ) für die Anpassung zur Laufzeit übersetzt (vergleiche Abbildung 2.10, Seite 20). Für das Experiment wurde eine konstante Versorgungsspannung von 1,5V genutzt.

Tabelle 4.5: Gemessene Stromaufnahme im Vergleich mit den zur Laufzeit ermittelten Werten unter Einbeziehung der Effizienz eines Spannungsreglers für das Fallbeispiel

Durchschnittliche Stromaufnahme in mA				
Gemessen	Effizienzadaption	Fixe Effizienz		
		schlechteste	beste	durchschnittliche
5,14	5,301	8,707	4,198	5,674

In Tabelle 4.5 sind die Ergebnisse des Experimentes dargestellt. Der Basisverbrauch des Spannungsreglers ist Teil der durchschnittlichen Stromaufnahme. Der Unterschied zwischen dem gemessenen Verbrauch und der Verbrauchserfassung mit Adaption an die Reglereffizienz beträgt 3%. Dieser Wert ist deutlich besser als die Nutzung eines statischen Effizienzwertes. Wird die beste gemessene Effizienz bei 1,5V Versorgungsspannung genutzt, wird der Verbrauch um 22% unterschätzt. Wird dagegen die schlechteste Effizienz angenommen, wird der tatsächliche Verbrauch um 68% überschätzt. Bei der Annahme einer durchschnittlichen Effizienz wird der Verbrauch immer noch um 10% überschätzt. Dieses einfache Experiment zeigt, dass eine Anpassung an die Effizienz eines Reglers besser als ein statisches Modell geeignet ist, den Verbrauch zu modellieren.

#### 4.4.5 Anwendungsbeispiel: Adaption des Anwendungsintervalls

Die gewonnenen Verbrauchsdaten können von einer Anwendung genutzt werden, um den eigenen Energieverbrauch zur Laufzeit zu steuern. Damit wird die Reaktion auf Veränderungen beispielsweise durch sinkende Versorgungsspannung möglich.

Die Informationen über den Verbrauch während der aktiven Phase wurden von einer Demonstrationsanwendung genutzt, um dynamisch die Frequenz der aktiven Phasen auf Grundlage der verfügbaren Energie und der gewünschten Laufzeit zu be-

rechnen. In jedem Intervall lauschte das System für 50ms auf Nachrichten, versendete eine Nachricht an die Basisstation und aktivierte eine LED für 200ms.

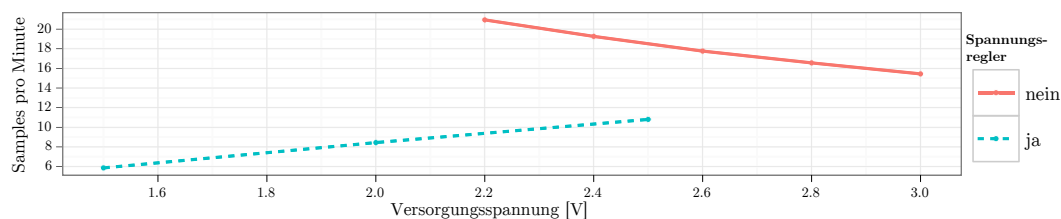


Abbildung 4.10: Dynamisch berechnete Anwendungsfrequenz für verschiedene Versorgungsspannungen

Abbildung 4.10 zeigt, wie die Anwendung ihr Intervall unter verschiedenen Versorgungsspannungen mit und ohne Spannungsregler festlegt. Obwohl der Regler die Nutzung kleinerer Versorgungsspannungen ermöglicht, verringert sich dabei die Frequenz, da die Effizienz des Reglers sinkt, aber die Versorgungsspannung der LED gleich bleibt. Ohne Regler ist die Frequenz höher, verringert sich aber mit steigender Spannung, da sich der Verbrauch der LED erhöht.

#### 4.4.6 Anwendungsbeispiel: Verbrauchserfassung eines Sensornetzes

Die durch den Software-basierten Ansatz ermöglichte Granularität der Verbrauchsdaten gestattet es, den Verbrauch von Anwendungen detailliert zu untersuchen. Um diese Möglichkeit zu demonstrieren, wurde der Energieverbrauch einer Sensornetzanwendung überwacht.

Zu diesem Zweck wurden sieben Knoten wie in Abbildung 4.11 dargestellt über mehrere Räume verteilt, zum Teil auf von den Decken hängende Leuchtstoffröhren.

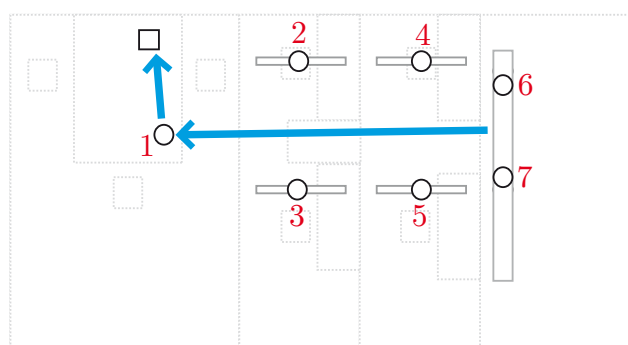


Abbildung 4.11: Der bei Experiment 6 verwendende Netzwerkaufbau: Nur Knoten 1 kann direkt mit der Senke kommunizieren.

Jedwede Kommunikation in Richtung Senke wurde über Knoten 1 abgewickelt, da nur dieser mit der Senke direkt kommunizieren konnte. Dies liegt in dem Umstand begründet, dass wie in [78] bereits beobachtet, der Accesspoint direkt über den



Rechner Energie bezog und damit eine höhere Spannung und Sendeleistung besaß als die per Batterie versorgten Knoten, was zu Problemen bei der Wegewahl führte. Es wurde eine einfache Kombination aus MAC und baumbasierter Wegewahl verwendet, die jede Minute das Funkmodul für vier Sekunden aktivierte. Innerhalb dieser Zeit fand die gesamte Kommunikation statt. Alle drei Minuten erzeugte die Anwendung auf den Knoten eine Nachricht, die über das Netz an den Rechner gesendet wurde. Basis für die Wegewahl war dabei die Entfernung zur Senke, welche in den ACK-Nachrichten übermittelt wurde. Jeder Knoten mit einer kleineren Entfernung leitete die Nachricht weiter, soweit er nicht eine Weiterleitung durch einen anderen Knoten observierte.

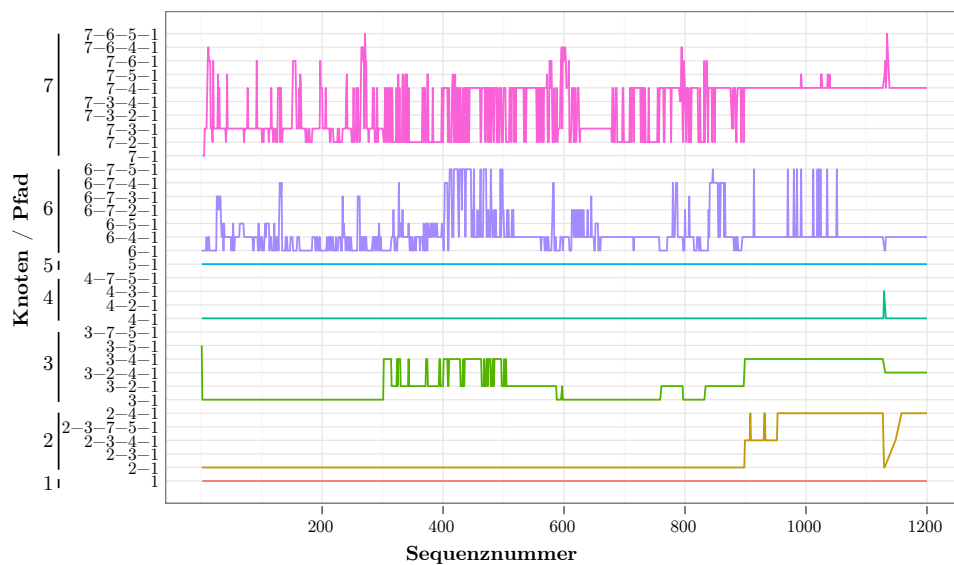


Abbildung 4.12: Wege der Nachrichten innerhalb von 60 Stunden nach Absender

Abbildung 4.12 stellt die Wege der Nachrichten innerhalb des Netzes für eine Laufzeit von 60 Stunden dar. Das Netz war nicht stabil, sondern veränderte sich. Nachrichten der weiter entfernten Knoten 6 und 7 wechselten häufig den Pfad. Der in der Mitte angesiedelte Knoten 5 konnte immer direkt mit der Senke kommunizieren, wohingegen die mit Knoten 1 benachbarten Knoten 2 und 3 zum Teil Umwege nutzten.

Jeder Knoten erfasste seinen Energieverbrauch mithilfe des vorgestellten Verfahrens und sendete diesen in der Nachricht an die Senke. Abbildung 4.13 schlüsselt den durch die Verbrauchserfassung ermittelten Verbrauch auf.

Wie zu erwarten hat Knoten 1 den höchsten Sendeverbrauch, gleichzeitig sinkt der Verbrauch für den Empfangsmodus. Dies ist in der kombinierten Send- und Empfangsphase begründet. Knoten 4 ist wie in Abbildung 4.12 zu erkennen Teil vieler Pfade, sodass er ebenfalls einen erhöhten Sendeverbrauch aufweist. Im Gegensatz dazu verbraucht Knoten 5, der kaum Nachrichten weiterleitet, weniger Energie für

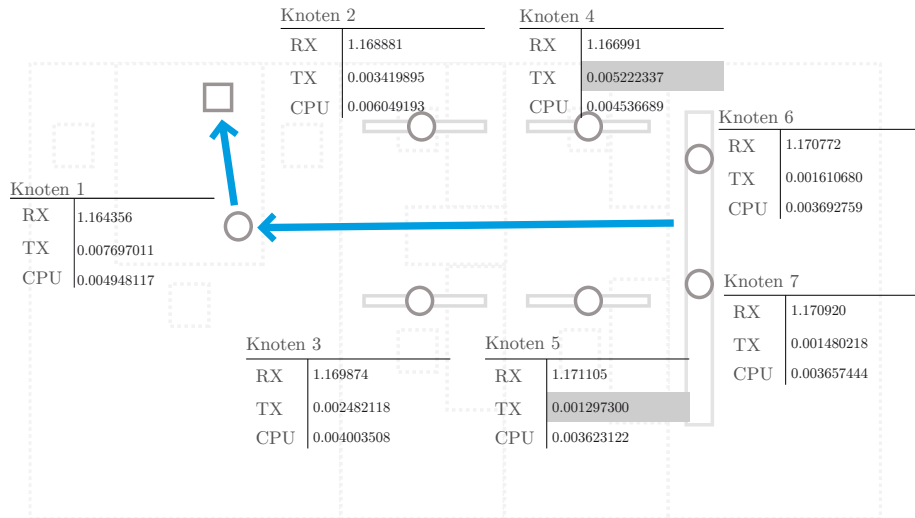


Abbildung 4.13: Ermittelte Stromaufnahme in mA nach 60 Stunden Laufzeit für jeden Knoten, aufgeteilt in CPU, Senden und Empfangen

das Senden und mehr für das Empfangen. Veränderungen in den Routen lassen sich beim Energieverbrauch für das Senden nachvollziehen, der in Abbildung 4.14 dargestellt ist. So steigt zum Beispiel der Verbrauch von Knoten 4 kontinuierlich nach der Anfangsphase.

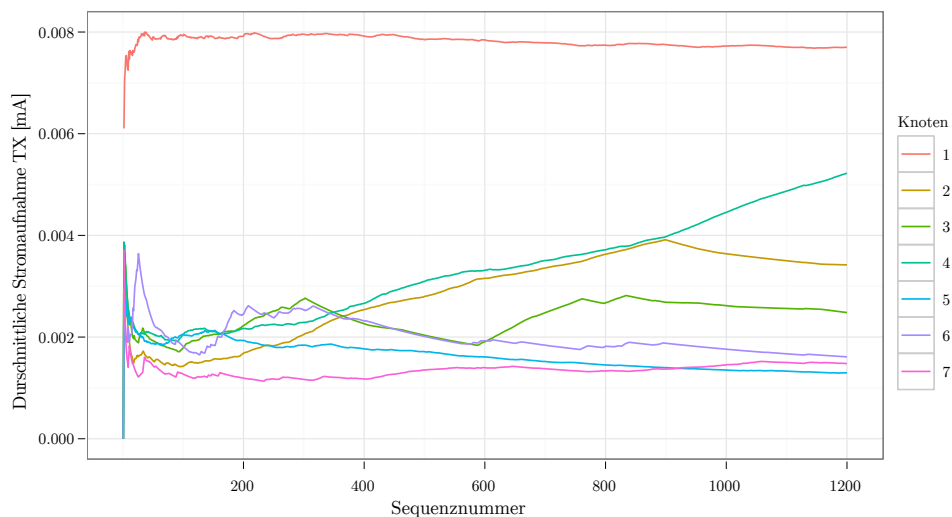


Abbildung 4.14: Durchschnittliche Stromaufnahme des Sendemodus für alle Knoten über 60 Stunden

## 4.5 Zusammenfassung

In diesem Kapitel wurde ein Verfahren für die Software-basierte Verbrauchserfassung zur Laufzeit vorgestellt, das im Gegensatz zu Hardware-basierten Ansätzen in der Lage ist, den Verbrauch feingranular auf einzelne Geräte und deren Modi aufzuschlüsseln. Diese Informationen ermöglichen neben der detaillierten Analyse des Anwendungsverhaltens eine Steuerung des Verhaltens einzelner Teile der Anwendung durch ein Energiemanagement. Der vorgestellte Ansatz ist unabhängig von der verwendeten Anwendung und kann ohne deren Anpassung in das Gesamtsystem integriert werden.

Es wurde gezeigt, wie die Verbrauchserfassung robuster gegen Variationen gemacht werden kann. Realistischere Modelle wurden auf Grundlage von Messungen erstellt, die einen Großteil der Knoten umfassen und nur zu geringer Überschätzung führen. Der Ansatz ist darauf ausgelegt, flexibel angepasst werden zu können, um sowohl an die Bedürfnisse in der Entwicklungsphase als auch an jene in der produktiven Ausbringung anpassbar zu sein. Im Gegensatz zu existierenden Ansätzen ist die präsentierte Verbrauchserfassung in der Lage, sich dynamisch an einen veränderlichen Verbrauch anzupassen. Dazu wurden Mechanismen vorgestellt, welche eine Anpassung an Spannungsveränderungen und die variable Effizienz eines Spannungsreglers ermöglichen. Diese lassen sich bei Bedarf auch erweitern, um auf andere Faktoren reagieren zu können, die den Energieverbrauch beeinflussen können, insbesondere die Temperatur.



## Kapitel 5

# Dynamische feingranulare Energiebudgets

Mithilfe der in den vorangegangenen Kapiteln vorgestellten Methoden lassen sich die verbleibende Energiemenge und der Verbrauch ermitteln. Dies ermöglicht es, die energetischen Ausgaben des Systems zu planen und zu steuern und somit im Voraus festgelegte Lebenszeitziele zu erreichen.

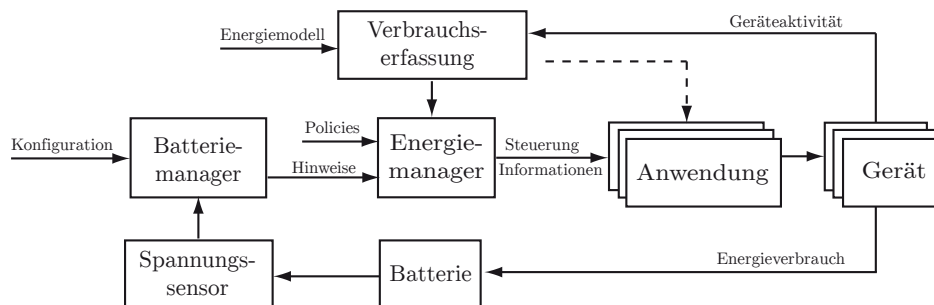


Abbildung 5.1: Steuerung individueller Anwendungsteile mithilfe eines generischen Energiemanagements auf Grundlage der verbrauchten und verfügbaren Energie

Abbildung 5.1 erweitert den bisherigen Ansatz (siehe Abbildungen 3.1 und 4.1 auf den Seiten 51 und 79) um ein generisches Energiemanagement. Dieses stellt Anpassungsmechanismen für Anwendungen bereit und vereinfacht so die Anwendungsentwicklung, da das Management nicht mehr individuell für jede Anwendung entwickelt werden muss. Eine feingranulare Steuerung wird durch eine Zuordnung der verbrauchten Energie zu einzelnen Anwendungsteilen möglich.

Grundsätzlich kann eine Steuerung auf verschiedene Arten erfolgen, bedarf aber in jedem Fall einer Einflussmöglichkeit auf die Anwendung. Neben einer direkten Einflussnahme des Managements auf Anwendungsparameter kann die Anwendung selbst mit Energiebewusstsein ausgestattet werden, um selbständig auf Vorgaben des Managements zu reagieren. Nicht alle Anwendungen haben dasselbe Interesse an

einem detaillierten Energiebewusstsein. Wie der Stand der Technik zeigt, existieren verschiedene Stufen des Energiebewusstseins. Anwendungen können verschiedene Kooperationsmöglichkeiten hinsichtlich Energie wahrnehmen, die idealerweise alle durch das Energiemanagement adressiert werden sollten.

In diesem Kapitel wird das Konzept der auf *Ressource Container* [24] basierenden Energiebudgets vorgestellt. Die Budgets stellen Informationen über die zur Verfügung stehende Energie in einem zeitlichen Kontext bereit und ermöglichen somit eine anwendungsseitige Anpassung. Die sich daraus ergebenden Möglichkeiten für das Management und die Anwendungen werden betrachtet.

## 5.1 Problembeschreibung

Auf der Basis der zur Verfügung stehenden Energiemenge  $E_{avail}$  und der verbleibenden Laufzeit  $t_{remain}$  ist es möglich, den Verbrauch der Anwendungen so zu beeinflussen, dass das Lebenszeitziel erreicht wird. Dies kann abhängig von der Anwendung verschiedenste Konsumenten betreffen, wie Geräte, Anwendungsteile und Ziele, die allesamt beachtet werden müssen.

Um jedem Konsumenten einen bestimmten Anteil  $\Phi$  von  $E_{avail}$  zu garantieren, ist es notwendig, sie energetisch voneinander zu isolieren. Eine Isolation hat jedoch den Nachteil, dass die Energieanteile möglicherweise nicht vollständig verbraucht werden können oder ein Energiebedarf unbefriedigt bleibt. Dies wird durch Dynamiken in Hard- und Software und dem daher nicht vollständig im Voraus berechenbaren Energiebedarf verursacht (siehe Kapitel 2.2).

So ist es möglich, dass der Bedarf vorher falsch eingeschätzt wurde oder ein kurzzeitiger Mehrbedarf aufgrund von Ausnahmesituationen besteht. Beides muss durch das Management behandelt werden können, um das Anwendungsziel nicht zu gefährden.

Die Veränderung von  $E_{avail}$  ist nicht nur vom Verbrauch des Systems abhängig, sondern kann von außen beeinflusst werden. Dies ist insbesondere dann der Fall, wenn die Batterieüberwachung die verbleibende Energie neu einschätzt. Daraufhin muss eine Anpassung der Anteile erfolgen, was wiederum Verteilungsprobleme generiert.

Abbildung 5.2 stellt diese Problematik schematisch dar. Eine einfache Sensor-netzanwendung wird in zwei Aufgabenbereiche (Kommunikation und Sensorik) aufgeteilt. Beide erhalten periodisch ihren Anteil ( $\Phi_{Funk}$  bzw.  $\Phi_{Sensor}$ ) an der zur Verfügung stehenden Energie, welche zu Beginn 28 Einheiten beträgt. Dabei wird darauf geachtet, die Minimalanforderungen zu erfüllen. Das heißt, eine Energieeinheit wird für die Aufrechterhaltung der Kommunikation bereitgestellt und zwei für die Datenerfassung. Die Verteilung der darüber hinaus verfügbaren Energie kann auf verschiedenste Arten erfolgen. In diesem Beispiel geht sie aufgrund höherer Priori-

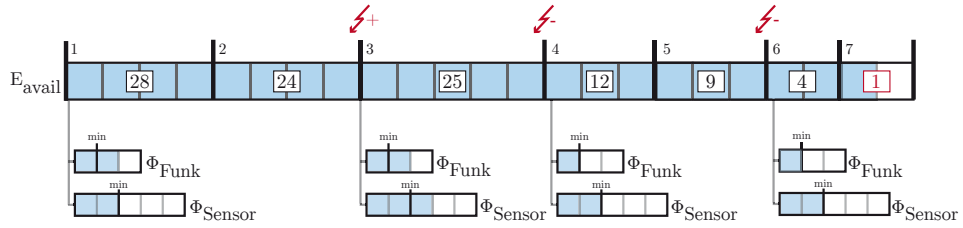


Abbildung 5.2: Schematische Darstellung des Verteilungsproblems für zwei Konsumenten bei dynamischer Energieverfügbarkeit: Die blauen Kästchen repräsentieren abstrakte Energieeinheiten. In den Phasen 3, 4 und 6 verändert sich die zur Verfügung stehende Energiemenge. Ab Phase 6 steht nicht mehr genug Energie zur Deckung des minimalen Bedarfs zur Verfügung.

tät der Kommunikation an  $\Phi_{Funk}$ . In Phase 3 ändert sich die zur Verfügung stehende Energie durch äußere Einflüsse, sodass mehr Energie auf die beiden Anwendungsbereiche verteilt werden kann. Dies kehrt sich in den Phasen 4 und 6 um, was dazu führt, dass in Phase 6 nicht mehr genug Energie verfügbar ist, um die Minimalanforderungen zu erfüllen. Eine solche Konfliktsituation kann auf verschiedene Arten aufgelöst werden. In diesem Fall werden beide minimale Anforderungen erfüllt, wodurch in Phase 7 nicht mehr ausreichend Energie für einen gesicherten Betrieb zur Verfügung steht.

Wie im Beispiel dargestellt, können die äußeren Einflüsse dazu führen, dass das System nicht mehr aus sich heraus in der Lage ist, mithilfe von sanfter Verschlechterung der Systemleistung das Lebenszeitziel zu erreichen, da beispielsweise die von der Anwendung vorgegebenen Parameter ein weiteres Absenken des Verbrauchs verhindern. In einem solchen Fall muss das System zwingend in der Lage sein, diese Ausnahmesituation anzuzeigen und dem Nutzer so die Möglichkeit zu geben, zu reagieren. Das kann beispielsweise durch eine vorgezogene Wartung vor Ort oder eine Parameteranpassung durch etwaige Fernwartung erfolgen.

## 5.2 Stand der Technik

Es existieren verschiedene Ansätze, um den Energieverbrauch eines drahtlosen tief eingebetteten Systems zu steuern. Traditionell ist das Ziel dabei eine Reduktion des Verbrauchs durch Energiesparmaßnahmen (siehe Kapitel 2.2). Im Gegensatz dazu soll bei einem zielorientierten Energiemanagement [23] eine definierte Laufzeit unter möglichst hoher Energieausnutzung erreicht werden. Eine Steuerung hat jedoch nur dann Sinn, wenn ausreichend Spielraum für das Management zur Verfügung steht, also mehr Energie als minimal nötig vorhanden ist, so dass das Ziel nicht nur knapp erreicht wird. Einige Ansätze wie [66] und [190] kommen aus dem Bereich der Energiegewinnung aus der Umwelt, da dort oftmals ein ähnliches Problem gelöst

werden musste: Trotz unsicheren Energieeinkommens muss das System durchgehend arbeiten, also auf Grundlage einer Prognose Energie verteilen und den Verbrauch beschränken. Dies entspricht einem zielorientierten Energiemanagement.

Alle Energiemanagementansätze nutzen jeweils eigene Lösungen, um die benötigten Informationen über die verbleibende Energiemenge zu erhalten. Sollen mehrere Anwendungsteile unabhängig voneinander gesteuert werden, ist eine korrekte Zuordnung des Verbrauchs zu diesen Teilen unerlässlich. Nicht in jedem Fall wird ein Bewusstsein über den Verbrauch des Systems benötigt, da sich dieser auch indirekt über die verbleibende Energie grob abschätzen lässt.

Grundsätzlich wird bei allen Verfahren die Leistung des Systems auf Grundlage einer oder mehrerer Steuergrößen verändert. Dies kann direkt durch das Management (Kapitel 5.2.1) oder indirekt (Kapitel 5.2.2) erfolgen.

Aufgrund existierender Varianzen in Hard- und Software (siehe Kapitel 2.2) muss ein Management in der Lage sein, auf einen veränderten Verbrauch in Ausnahmesituationen zu reagieren. Ein veränderter Verbrauch kann entweder kurzfristig auftreten oder von längerer Dauer sein. Idealerweise kann ein kurzfristiger Mehrbedarf beispielsweise aufgrund von Phänomendetektion oder Retransmissionen gedeckt werden. Ein kurzfristiger Minderbedarf sollte nicht zu einer sofortigen Anpassung und damit Auswirkung auf andere Anwendungsteile führen. Zusätzlich sollte das System in der Lage sein, die zugeteilten Energiemengen an den tatsächlichen Bedarf anzupassen, um sowohl eine Gefährdung des Anwendungsziels als auch Energieverschwendung zu vermeiden.

### 5.2.1 Direkte Steuerung

Bei einer direkten Steuerung nutzt das Energiemanagement Regler der Anwendungen beziehungsweise Treiber, um Parameter oder Ausführungspfade zu verändern. Die Anpassung kann direkt erfolgen oder durch die Anwendung vorgenommen werden, die auf Anfragen des Managements reagiert. Die Anwendungen treffen jedoch keine eigenständigen Entscheidungen, sondern konsultieren immer das System.

#### Odyssey

Im Unix basierten Odyssey [191] wird die Genauigkeit/Wiedergabetreue von Applikationen angepasst, um die vom Nutzer gewünschte Laufzeit bei maximaler Leistung zu erreichen. Zu diesem Zweck wird der durch eine Hardwarelösung erfasste Verbrauch genutzt, um auf Basis historischer Daten den zukünftigen Verbrauch vorherzusagen. Weicht die aus initial bekannter Energiemenge und zukünftigem Verbrauch errechnete Laufzeit vom Ziel ab, muss sich die laufende Applikation anpassen. Adaptionsmöglichkeiten können direkt durch Modifikation der Anwendungen oder indirekt über Stellvertreter geschaffen werden, die die Ein- oder Ausgabe der



Anwendungen beeinflussen. Bei nebenläufigen Anwendungen erfolgt die Anpassung über durch den Nutzer vergebene Prioritäten.

Der Ansatz zielt darauf ab, die Zahl der Anpassungen durch Mittelwertbildung möglichst gering zu halten, wodurch kurzzeitige Varianzen abgefangen werden können. Langfristige Bedarfsänderungen sind allerdings nicht direkt behandelbar, da das System immer nur gerade laufende Anwendungen und deren Verbrauch betrachtet. Dadurch besteht keine vollständige Isolation zwischen den laufenden und pausierten Anwendungen, wodurch der Mehrbedarf einer laufenden Anwendung andere benachteiligen und im Extremfall verhungern lassen kann.

### **Imprecise Computations**

Imprecise Computations [192] stammt ursprünglich aus dem Scheduling-Kontext und dient zur Einhaltung von Zeitanforderungen von Echtzeittasks. Jede Task wird in eine notwendige und eine optionale Subtask aufgeteilt. Optionale Subtasks werden nur dann ausgeführt, wenn keine Zeitanforderungen von notwendigen Subtasks behindert werden. In EPOS [67] wird dieses Konzept auf das Erreichen von Lebenszeitzielen ausgeweitet. Optionale Subtasks werden nur bei ausreichender Energieverfügbarkeit ausgeführt. Diese wird periodisch mittels Überwachung des Spannungsabfalls der Batterie im Zusammenspiel mit vordefinierten Verbrauchswerten errechnet.

EPOS bietet keine Isolation zwischen einzelnen Tasks. Die Entscheidung, ob eine optionale Subtask ausgeführt werden kann, wird rein auf Basis der globalen Energieverfügbarkeit vom Scheduler getroffen. Da die Ermittlung der verbleibenden Energie nur periodisch erfolgt, haben kurzfristige Veränderungen des Energiebedarfs keine direkte Auswirkung. Langfristige Bedarfsänderungen können nicht direkt behandelt werden. Insbesondere ein dauerhafter Mehrbedarf kann nicht befriedigt werden, da eine Zuordnungsmöglichkeit fehlt.

### **Energy Levels**

Die in [21] vorgestellte Programmierabstraktion der *Energy Levels* verfolgt ein ähnliches Prinzip wie Imprecise Computations, zielt aber explizit auf Sensorknoten. Um ein Lebenszeitziel zu erreichen, wird zwischen verschiedenen vom Nutzer definierten globalen Anwendungsstufen gewechselt und so der Verbrauch gesteuert. Diese Stufen stellen unterschiedliche Funktionalität bei unterschiedlichem Verbrauch bereit, wobei eine höhere Stufe einen höheren Energiebedarf hat. Um dem System die Stufenwahl zu ermöglichen, muss jede Stufe mit einem Nutzen-Wert versehen werden. Der Verbrauch jeder Stufe muss statisch im Voraus ermittelt werden, beispielsweise durch Simulation. Neben dem einmaligen Verbrauch durch die Programmausführung wird auch der kontinuierliche Verbrauch durch Zustandsänderungen der Hardware einbezogen. Zur Laufzeit wird die verbrauchte Energie jeder Stufe protokolliert. Die

in der Batterie zur Verfügung stehende Energie wird durch einen tabellenbasierten Ansatz (vgl. Kapitel 3.2.3 Seite 57) ermittelt. Auf dieser Basis wird für jede Stufe die mögliche Laufzeit berechnet und anschließend durch Optimierung eine Stufenbelegung mit maximalem Nutzen für die verbleibende Laufzeit berechnet. Um den Speicheraufwand zu verringern und den Nutzen zu maximieren, werden nach jeder Berechnung nur die höchst mögliche Stufe gewählt und alle weiteren Daten verworfen. Dieses Verhalten ist optimistisch, da nicht von einer zukünftigen Verringerung der verbleibenden Energie ausgegangen wird. Um dies auszugleichen, wird ein dynamisch sinkender Sicherheitsfaktor in die Ermittlung der verbleibenden Energie einbezogen, sodass das Lebenszeitziel auch bei ungenauer Information über Verbrauch und verbleibende Energie erreicht werden kann. Durch diesen Faktor steigt im Laufe der Zeit, sofern die Annahmen über den Verbrauch und die verfügbare Energie korrekt sind, die der Anwendung zur Verfügung stehende Energiemenge und somit potenziell die Anwendungsstufe.

Durch die optimistische Zuweisung der Anwendungsstufen ist die Dimensionierung des Sicherheitsfaktors zentral für das Erreichen des Ziels. Da die Verbrauchserfassung nur auf Granularität der Anwendungsstufen stattfindet, wirkt sich eine Bedarfsänderung, sofern durch die statische Erfassung observiert, auf alle in dieser Stufe zusammengefassten Anwendungsteile aus. Der Mehrbedarf eines Anwendungsteils kann nicht beschränkt werden, sondern führt potenziell zur Verringerung der Anwendungsstufe und betrifft somit alle Anwendungsteile. Von einem Minderbedarf profitiert im Gegenzug die gesamte Anwendung, denn er kann zu einer Stufenerhöhung führen. Da das System die Energiestufen periodisch neu bestimmt, wirken sich kurzfristige Bedarfsänderungen nicht unmittelbar aus.

## Eon

Der mit Eon [66] vorgestellte Ansatz versucht, einen dauerhaften Betrieb durch Balancierung von Energieeinkünften - primär durch Solar - und Verbrauch zu erreichen. Er basiert auf einer Koordinierungssprache und beschreibt den Fluss von Daten durch die verschiedenen Komponenten. Für eine Anpassung des Verbrauchs auf Systemebene können abstrakte, geordnete Energiestufen für die Komponenten definiert werden. Das Laufzeitsystem bestimmt die aktuelle Stufe. Komponenten mit dieser oder einer kleineren Stufe werden genutzt. Zusätzlich werden für eine Feinanpassung adaptive Uhren eingesetzt, deren Frequenz vom Laufzeitsystem innerhalb eines vorgegebenen Bereiches bestimmt wird. Der Verbrauch jeder Komponente wird zur Laufzeit durch Hardware bestimmt.

Zwar wird der Verbrauch auf der Ebene von Komponenten und Datenpfaden der Anwendung erfasst. Da diese aber in den abstrakten Energiestufen zusammengefasst werden, kann sich die Bedarfsänderung eines Anwendungsteils direkt auf andere

auswirken. Kurzzeitige Varianzen im Verbrauch wirken sich durch die periodische Anpassung nicht direkt aus, sondern können möglicherweise relativiert werden.

### **IDEA**

IDEA [193] verfolgt den Ansatz, den Energieverbrauch innerhalb einer Knotennachbarschaft zu steuern, indem der Einfluss lokaler Anpassungen auf den Energieverbrauch benachbarter Knoten in die Adaptionentscheidungen einbezogen wird. Dazu werden der durch ein Zustandsmodell abgeschätzte Verbrauch, der Batterieladestand und der Energiegewinn durch Harvesting erfasst, modelliert und verbreitet. Applikationen stellen verschiedene Zustände bereit, die sich in Nutzen und energetischer Auswirkung auf benachbarte Knoten unterscheiden. Diese Informationen müssen dem System von der Anwendung bereitgestellt werden. Auf Grundlage dieser Daten wird durch eine Zielfunktion der optimale Zustand bestimmt. Die Zielfunktion kann unterschiedliche Verhaltensweisen moderieren. Der Ansatz nennt als Beispiele die Maximierung der durch Harvesting gewonnenen Energie und die Maximierung der Laufzeit aller Knoten. Zwar können diese Ziele global optimiert werden, aber gerade die notwendigen Eingabedaten machen das System aufwendig und erlauben bisher nur die Verwendung einer adaptiven Anwendung. Mangels paralleler Anwendungsteile beziehungsweise Ziele ist somit keine Isolation notwendig. Das Verhalten bei Bedarfsänderungen gleicht durch die Verwendung von Anwendungsstufen dem von Energy Levels und Eon.

### **Energy Management Architecture**

Die Energy Management Architecture (EMA) [194] besteht aus drei Teilen: einer Regelschnittstelle, einem Energiemonitor und dem Energiemanager. EMA separiert den Sensornetz-Nutzer vom Entwickler. Der Entwickler schreibt und annotiert die Softwarekomponenten. Der Nutzer bestimmt das Verhalten der Anwendung durch die Regelschnittstelle.

Mithilfe der Regelschnittstelle kann der Nutzer priorisierte Ziele formulieren, wie beispielsweise: „Erreiche eine bestimmte Laufzeit und maximiere dabei die Datenerfassung“. Regeln beziehen sich auf Transaktionen, welche in EMA abgeschlossene, Energie verbrauchende Handlungen bezeichnen. Die möglichen Regeln beschränken sich auf Ausführungsintervalle und Zielvorgaben, jedoch lassen sich Regeln mittels logischer Schlüsselwörter (AND, OR) verbinden. Zudem ist es möglich, Regeln mit Maximierung oder Minimierung zu nutzen, wobei nur eine solche Regel im System zugelassen ist. Der Energiemonitor erfasst den Verbrauch jeder Transaktion durch Hardware. Um die Zuordnung des gemessenen Verbrauchs zu Transaktionen zu ermöglichen, muss der gesamte Code mit Systemaufrufen annotiert werden, die den Beginn und das Ende einer bestimmten Transaktion festlegen. Zusätzlich können

Transaktionen nicht nebenläufig ausgeführt werden. Die in einem Baum übersetzten Regeln werden vom Energiemanager beim Beginn jeder Transaktion ausgewertet und auf Basis des historischen Verbrauchs angewendet.

Durch die kontinuierliche Auswertung der Regeln können sich auch kurzzeitige Varianzen im Energieverbrauch auswirken. Ein langfristiger Mehrbedarf ist nur für priorisierte Transaktionen möglich, da sich weniger wichtige Transaktionen in einem solchen Fall einschränken müssen. Das System profitiert jedoch insgesamt von einem dauerhaften Minderbedarf einzelner Teile.

### **Variability-aware Software Adaption**

In [195] wird ein Management präsentiert, das die Intervalle der Anwendung aufgrund des temperaturabhängigen Verbrauchs während des Schlafzustandes anpasst. Zu diesem Zweck muss für jeden Knoten ein temperaturabhängiges Verbrauchsprofil erstellt werden. Das optimale Verhältnis zwischen aktiver Zeit und Schlaf kann dann statisch auf Grundlage von dem temperaturabhängigen Verbrauchsprofil des Knotens, dem Temperaturprofil des Ortes, dem Lebenszeitziel und der Batteriekapazität berechnet werden. Ein dynamischer Ansatz unter Einbeziehung der tatsächlich verbrauchten Energie wird nur skizziert und vorgeschlagen, das exakte Temperaturprofil des Ausbringungsortes von den Knoten zur Laufzeit autonom lernen zu lassen. Der Nutzer spezifiziert für jeden Task eine Lebenszeit und die akzeptierte Spanne der Anwendungsintervalle. Das System teilt die mögliche aktive Zeit unter den Tasks auf. Die Intervalle der Tasks werden anschließend mithilfe der von den Tasks zur Ausführung benötigten Zeit durch das adaptive Scheduling berechnet. Hierfür wird eine einfache, auf der Ebene der einzelnen Tasks umgesetzte Zeiterfassung durch Zeitstempel genutzt, wobei andere Verbraucher nicht einbezogen werden. Varianzen in der Energieversorgung werden ebenfalls nicht einbezogen.

Durch die taskweise Erfassung und daraus folgende Berechnung des Intervalls wirken sich Bedarfsänderungen nur auf die jeweiligen Tasks aus. Da die Verteilung der aktiven Zeit auf die Tasks fix ist, können andere Tasks nicht von einem etwaigen Minderbedarf profitieren oder sich bei einem Mehrbedarf einschränken. Unklar ist das Verhalten der Tasks, wenn die angegebenen Spannen infolge des Verbrauchs verlassen würden. Ist keine Anpassung darüber hinaus möglich, wird durch die Isolation im besten Falle Energie verschwendet. Im schlechtesten Fall jedoch ist das Lebenszeitziel durch einen zu großen Verbrauch in Gefahr.

### **PControl**

In [23] wird der Energieverbrauch der Anwendung durch die Dauer von Schlafphasen gesteuert. Als Basis dient der in Kapitel 3.2.3 beschriebene Ansatz der Spannungsvorhersage. Weicht die gemessene Spannung von der erwarteten ab, wird die Dauer

der Schlafphase verändert. Die Dauer wird durch Lösen eines Optimierungsproblems bestimmt.

Bei diesem Ansatz ist keine direkte Erfassung des Verbrauchs notwendig, da nur eine Anwendung auf dem System läuft und der Spannungsabfall als Steuergröße das Intervall bestimmt. Änderungen des Verbrauchs werden somit nur indirekt erfasst und durch Anpassung des Intervalls direkt ausgeglichen.

### 5.2.2 Indirekte Steuerung

Bei der indirekten Steuerung treffen die Anwendung beziehungsweise unabhängige Teile davon eigenständig energierelevante Entscheidungen. Das Management gibt dabei den Rahmen vor, beispielsweise durch Zuteilung von Energie zu einzelnen Anwendungsteilen, auf deren Verfügbarkeit die Anpassungsentscheidungen der Anwendung basieren. Zur Speicherung der Information über den Ressourcenverbrauch von (Sub-)Tasks wird oftmals auf das Konzept der *Ressource Container* (RC) [24] zurückgegriffen. Unterarten speziell für den eingebetteten Bereich und mit einem Fokus auf die Bereitstellung von detaillierten Energieverbrauchsinformationen sind beispielsweise *Energy Capsules* [185] und *Energy Container* [196]. Beide dienen jedoch nur der Informationssammlung, nicht dem Energiemanagement.

### Virtual Battery

*Virtual Battery* [197] ermöglicht eine Isolation hinsichtlich der Energie von parallelen Anwendungen auf einem Sensorknoten. Die Energie wird auf alle Anwendungen statisch aufgeteilt. Jede Anwendung kann ihre Energie unabhängig von anderen verwalten, so als würde sie allein auf dem Knoten laufen. Um diese Sicht zu ermöglichen, werden alle Geräte ebenfalls virtualisiert. Aufrufe, die die Gerätezustände ändern, sind aus Sicht der einzelnen Anwendung immer erfolgreich und verändern den Energieverbrauch, werden aber nur dann tatsächlich ausgeführt, wenn dies keine Konflikte verursacht. Jede Applikation reserviert sich einen Prozentsatz der physischen Batteriekapazität und gibt dabei eine Laufzeit und ein Auszahlungsintervall an. Letzteres ermöglicht es einer Anwendung, sich selbst zu limitieren. Nicht genutzte Energie kann angespart und auf einmal verwendet werden. Zusätzlich besteht für die Anwendung die Möglichkeit, einen kurzfristigen Mehrbedarf durch einen Kredit zu decken, definiert als Prozentsatz aus den eigenen zukünftigen Auszahlungen. Eine weitere Selbstlimitierung ist durch das Festlegen einer maximalen Verbrauchsrate möglich, welche die maximal konsumierbare Energie innerhalb eines angegebenen Zeitintervalls definiert. Ist diese Verbrauchsrate überschritten oder die private Energie aufgebraucht, wird die Anwendung suspendiert.

Der Ansatz weist in der vorgestellten Form mehrere Schwachstellen auf. Erstens ist die Virtualisierung nicht fair im Sinne von [196]. Die Kosten einer geteilten

Ressource werden nicht geteilt, sondern nur von einer Anwendung getragen, alle anderen Nutzer haben keine Kosten. Zweitens kennt das System keine Dynamik. Die Betrachtung von Sonderfällen beschränkt sich auf einen kurzfristigen Mehrbedarf. Ein konstanter Mehrbedarf wird nicht behandelt, da das Reservierungsverhältnis zu Beginn festgelegt wird. Des Weiteren kann ungenutzte Energie nicht verwendet werden. Änderungen in der tatsächlich vorhandenen Energiemenge werden nicht betrachtet. Beispielsweise wäre eine Anpassung zur Laufzeit durch die Möglichkeit der einmaligen Auszahlung des gesamten Anteils potenziell unfair, da die Energie bereits am Anfang aufgebraucht werden kann.

### Self-Organizing Resource Allocation

Beim marktorientierten Ansatz der Self-Organizing Resource Allocation (SORA) [198] werden Knoten in einem Fahrzeugverfolgungsszenario als individuelle Agenten betrachtet, die ihren Profit maximieren wollen. Diesen erhalten Knoten für das Durchführen von sinnvollen, das heißt dem Netzwerkziel dienenden, lokalen Aktionen als Reaktion auf global ausgeschriebene Preisinformationen. Die Energiekosten jeder Aktion sind bekannt, zusätzlich steht jedem Knoten nur eine begrenzte, aber bekannte Energiemenge zur Verfügung. Die Begrenzung ermöglicht das Erreichen einer definierten Lebenszeit. Um zu verhindern, dass Knoten ihr Budget durch teure Aktionen mit hohem Profit gleich am Anfang aufbrauchen und somit kostengünstige Aktionen nur bei niedrigem Energiestand durchführen, ist das Budget als *Token Bucket* implementiert, in das kontinuierlich Energie gefüllt wird. Das Fassungsvermögen des Buckets bestimmt die auf einmal nutzbare Energie. Ist das Bucket leer, muss der Knoten einen Schlafzustand einnehmen, bis es wieder ausreichend nachgefüllt wurde. Jeder Knoten lernt durch verstärkendes Lernen anhand vorangegangener Zahlungen, welche Aktionen profitabel für ihn sind. Das gewünschte Netzwerkverhalten wird durch die Preisgestaltung angeregt, wobei Änderungen an alle Knoten propagiert werden müssen. Für die Preisgestaltung wird die Nutzung von Experimenten in Testbeds empfohlen, da das komplexe Agentenverhalten in Kombination mit der unvorhersehbaren Umwelt problematisch ist. So können kleine Änderungen große Auswirkungen haben beziehungsweise selbst große Veränderungen nicht das gewünschte Verhalten erzielen.

Durch das *Token Bucket* werden die Anwendungsteile nicht voneinander isoliert, gleichzeitig ist der Verbrauch durch die verfügbaren Token beschränkt, sodass möglicherweise lohnende Aktionen mangels Token nicht ausgeführt werden können. Zusätzlich können Token beim Erreichen des Fassungsvermögens verloren gehen, dann wird Energie verschwendet. Veränderungen im Verbrauch einzelner Aktionen sind nicht observierbar, da von einem bekannten Verbrauch ausgegangen und gleichzeitig der Batteriestand nicht einbezogen wird.

### Cinder

Das Energiemanagement bei Cinder [199] zielt auf Mobiltelefone mit komplexen nebenläufigen Applikationen ab. Energie wird in Reserven gespeichert, welche auf *Ressource Containern* basieren. Reserven werden mittels sogenannter *taps* verbunden, sodass ein hierarchischer Aufbau ausgehend von einer Basisreserve erfolgt. *Taps* steuern den Energiefluss von einer Reserve in die andere, wobei der Fluss, ähnlich wie bei einem *Token Bucket*, pro Zeiteinheit begrenzt ist und somit das Erreichen von Laufzeitzielen möglich wird. Der Energieverbrauch wird durch ein Zustandsmodell erfasst, dessen Werte im Voraus ermittelt werden. Durch einen Prozess verursachter Verbrauch wird aus der assoziierten Reserve entfernt, wobei die Kosten geteilter Ressourcen auf alle Nutzer aufgeteilt werden. Reserven ermöglichen drei Prinzipien: Die Isolation beschränkt den Verbrauch von Prozessen. Bei der Ablaufplanung wird geprüft, ob Energie in einer dem Thread zugeordneten Reserve vorhanden ist, andernfalls wird dieser nicht ausgeführt. Die Delegation ermöglicht es, die Kontrolle über einen Teil der in der Reserve gespeicherten Energie zu teilen. Auf diese Weise wird eine Kooperation ermöglicht, beispielsweise durch Bildung eines Pools zum Bezahlen kostenintensiver Geräteaktionen. Die Unterteilung und Bildung von Subreserven ermöglicht es, die Kontrolle über einen Teil der Energie vollständig abzugeben. So kann ein Prozess bei der Erzeugung von Kindprozessen individuelle Reserven aus seiner eigenen Reserve ableiten.

Eine Rückgewinnung ungenutzter Energie erfolgt durch *backward proportional taps*. Dabei wird, ab einer definierten Energiemenge, die in der Reserve gespeichert ist, ein prozentualer Anteil zurückgeführt. Um weiteres Anhäufen von Energie zu verhindern, verlieren alle Reserven 50% ihrer gespeicherten Energie nach einer längeren Frist.

Durch den kontinuierlichen Energiefluss zwischen den Reserven entsteht ein beträchtlicher Aufwand an Scheduling und Kontextwechseln, der im Ansatz durch periodisches Ausführen verringert wird. Zusätzlich ist unklar, wie Cinder bei Veränderungen in der Energieverfügbarkeit reagiert, da die Transferraten durch die *Taps* vorgegeben sind. Dadurch kann ein kurzzeitiger Mehrbedarf auch nur von angesparter Energie gedeckt werden.

### ECOSystem

Der von ECOSystem [69] verfolgte Ansatz ist die Limitierung der durchschnittlichen Last auf die Batterie, um die Laufzeit zu verlängern und nutzerdefinierte Laufzeitziele zu erreichen. Dazu wurde das sogenannte *Currency*-Modell eingeführt, welches das Recht auf Energie innerhalb eines Zeitlimits repräsentiert. *Ressource Container* werden genutzt, um zum einen den Verbrauch von zugeordneten Anwendungen oder Tasks (eine Task kann aus mehreren Prozessen bestehen und beschreibt eine Auf-



gabe) zu erfassen und zum anderen die *Currency* zu erhalten und zu speichern. Der Verbrauch wird durch ein zweiteiliges Software-Modell erfasst. Die Basiskosten umfassen Schlafzustände und den Verbrauch von Standardzuständen von Geräten, die nicht in das Modell integriert sind. Die explizit verwalteten Geräte erfassen den Verbrauch auf der Ebene der Zustände. Reine Basiskosten werden nicht aus *Ressource Containern* beglichen.

Die Laufzeit ist in Epochen aufgeteilt, an deren Beginn die verfügbare Menge *Currency* durch die gewünschte Last mithilfe einer Batterieformel ermittelt wird. Anschließend wird die *Currency* auf Grundlage von durch den Nutzer festgelegten prozentualen Verhältnissen auf die konkurrierenden Tasks verteilt.

Die initiale Management Regel wird als pay-as-you-go bezeichnet. Dabei wird für die Verwendung von Geräten *Currency* aus dem Container der Task abgezogen. Ist diese aufgebraucht, wird die Verarbeitung gestoppt. Die potenzielle Folge ist, dass alle Tasks ihre *Currency* so schnell wie möglich innerhalb einer Epoche aufbrauchen.

Bei asynchronen Gerätezugriffen, wie beispielsweise beim Festplattenzugriff, wird der zugehörige Resource Container mit den Daten gepuffert und erst nach der erfolgten Hardwareaktion belastet. Kosten, die aufgrund von Geräte(de)aktivierung oder durch Leerlauf entstehen, werden periodisch oder am Ende der Operation anteilig auf die nutzenden Container verteilt. Durch diese Asynchronität ist es möglich, dass ein Container keine *Currency* mehr enthält und sich verschuldet.

Nicht genutzte *Currency* kann bis zu einem Limit angespart werden, alle weitere *Currency* wird verworfen, um ein exzessives Ansparen zu verhindern. Dies hat zur Folge, dass weniger Energie als erwartet verbraucht wird und die Rückkopplung mit der Batterie durch das SMART-Interface die Menge der verfügbaren *Currency* langsam erhöht. In [200] wird diese Vorgehensweise zugunsten einer *Currency* erhaltenden Verteilung verändert. Bei dieser erfolgt die Verteilung von *Currency* in zwei Schritten. Zunächst werden die Limits aller Container aufgrund des historischen Verbrauchs angepasst. Anschließend wird die *Currency* verteilt, wobei überzählige *Currency* anteilig auf andere Container verteilt wird. Erst wenn alle gefüllt sind, geht *Currency* verloren. Der Scheduler wird durch einen energiezentrischen Planer ersetzt, der die Ausführung von Tasks auf die gesamte Epoche verteilt und so ein stoßweises Aufbrauchen der Energie am Anfang einer Epoche verhindert.

Problematisch ist die Behandlung von Ausnahmesituationen durch fehlende Dynamik zwischen den Containern. Der tatsächliche Bedarf spiegelt sich nur an den Limits wider, welche letztlich nur dann benötigt werden, wenn der Bedarf kleiner als erwartet ist. Der umgekehrte Fall, dass die Energie kurzfristig nicht ausreicht, kann nicht innerhalb einer Epoche behandelt werden. Auch die Anpassung zwischen den Epochen erfolgt letztlich nur, wenn der Bedarf insgesamt kleiner als die zur Verfügung stehende *Currency*-Menge ist.



### Pixie

Pixie [201] ist ein Betriebssystem für datenintensive Sensornetze mit dementsprechend hohen Datenraten. Es basiert auf einem Datenflussprogrammiermodell, bei dem Verarbeitungsstufen miteinander verbunden werden. Das Ressourcenmanagement nutzt sogenannte *Ressource Tickets*. Diese repräsentieren die Verfügbarkeit und zeitgebundene Reservierung von Ressourcen, vergleichbar mit ECOSystems *Currency*, und werden von sogenannten Brokern verwaltet, welche die Details und Regeln der Ressourcenvergabe umsetzen.

Ressourcen selbst werden durch Allokatoren verwaltet. Diese bestimmen die Verfügbarkeit einer Ressource und allozieren auf Anfrage Tickets, ohne jedoch Regeln aufzuerlegen. Der Energie Allokator schätzt die verbleibende Laufzeit mithilfe eines softwarebasierten Zustandsmodells ab, wobei die CPU zur Verringerung des Aufwands nur pauschal erfasst wird. Eine dynamische Einbeziehung der Batterie findet nicht statt.

Der Energie Broker vergibt Tickets auf Grundlage einer Entladerate, die sich aus der gewünschten Laufzeit und der verbleibenden Energie bildet. Stufen registrieren ihren notwendigen Bedarf für eine Arbeitseinheit zusammen mit einer Priorität. Wie der Bedarf ermittelt wird, wird nicht näher spezifiziert und auf statische oder dynamische Techniken verwiesen.

Ein energiebewusster Schalter verteilt eingehende Daten aufgrund der Energieverfügbarkeit auf verschiedene Ausgänge. Ein energiebewusster Filter gibt nur dann Daten weiter, wenn ausreichend Energie vorhanden ist, und steuert somit den Verbrauch von nachfolgenden Stufen. Eine Kopplung von Daten und einem für die Verarbeitung notwendigen *Ressource Ticket* ist für die meisten Anwendungen sinnvoll und gestattet es, beide durch die Stufen fließen zu lassen.

Eine Kreditregel lässt Applikationen einen kurzfristigen Mehrbedarf decken, selbst wenn die Entladerate verletzt würde. Nach Überschreiten eines Limits werden jedoch so lange keine Tickets mehr generiert, bis die Rate wieder ausgeglichen ist. Ein kurzfristiger Minderbedarf führt nach Ablauf des Tickets zu einer Neuvergabe der Energie in einem anderen Ticket.

### Nemesis

Das Betriebssystem Nemesis [68] nutzt mikroökonomische Mechanismen, um den Verbrauch von Anwendungen zu steuern. Anwendungen müssen für die Nutzung von Ressourcen bezahlen, wobei sie entsprechend ihrer Priorität einen Anteil an der Währungsmenge erhalten. Ein Schattenpreis dient als Rückkopplung für die Anwendungen und wird anhand des Verbrauchs dynamisch bestimmt. Dieser steigt, wenn der Verbrauch zu groß wird, und sinkt, wenn ein höherer Verbrauch möglich ist. Zusätzlich müssen Anwendungen, die einen hohen Anteil am Gesamtverbrauch haben,

proportional mehr zahlen. Anwendungen versuchen in Nemesis, ihren vom Anwender definierten verbrauchsabhängigen Nutzen zu maximieren und dabei die Kosten zu minimieren. Der Verbrauch wird prozessweise mithilfe eines Smart Battery Systems ermittelt, wobei die Anteile der Geräte durch ein kalibriertes Modell ermittelt werden. Diese Informationen werden jedem Prozess zur Verfügung gestellt.

Durch das Schattenpreismodell ist eine langfristige Planung für die Anwendungen schwierig, weil sich der Preis unabhängig von ihrem Verhalten ändern kann und damit die energetische Isolation zwischen den Anwendungen wieder aufgebrochen wird. Dies ist insbesondere problematisch, wenn einzelne Anwendungen zwangsweise einen hohen Energiebedarf haben, selbst aber nur wenig auf Veränderungen reagieren können, da das System keine Einsicht in die Anforderungen der Anwendung hat und keine Umverteilung der Währung möglich ist. Ein Mehr- oder Minderbedarf wirkt sich aufgrund der daraus folgenden Preisänderung sofort auf diesen und andere Prozesse aus. Durch den festgeschriebenen Anteil und die proportionale Preisermittlung kann ein dauerhafter Mehrbedarf nicht befriedigt werden.

### 5.2.3 Zusammenfassung der Ansätze

Tabelle 5.1 fasst die betrachteten Ansätze zur Steuerung des Energieverbrauchs, um Lebenszeitziele zu erreichen, zusammen und bewertet sie. Die Bewertung erfolgt dabei nach einem mehrteiligen Maßstab: sehr gut (++), gut (+), neutral (o), schlecht (-) und sehr schlecht (--).

Für das Erreichen von Lebenszeitzielen ist es notwendig, dass das Management auf Bedarfsänderungen reagieren kann. Solche Änderungen können direkt durch Anforderungen der Anwendung oder indirekt durch einen veränderten Verbrauch der Hardware ausgelöst werden. Die Art der Reaktion ist durch die Ansätze unterschiedlich umgesetzt worden. Ein kurzfristiger Mehrbedarf kann durch Kredite behandelt werden (Virtual Battery [197], Pixie OS [201]). Der Einfluss eines Minderbedarfs kann durch Mittelwertbildung bei der Verbrauchserfassung und eine periodische Anpassung der Parameter verringert werden. Eine langfristige beziehungsweise dauerhafte Anpassung setzt voraus, dass sich das Verhältnis ändern lässt, in dem Anwendungsteile Energie erhalten. Während dies bei der direkten Steuerung meistens impliziert erfolgen kann, setzt eine indirekte Steuerung weitere Maßnahmen voraus. Energie sollte zwischen einzelnen Anwendungen oder deren Teilen verschoben werden können. Zusätzlich muss ein unbeschränktes Ansparen verhindert werden.

Die Flexibilität der Ansätze hängt von der zugrundeliegenden Steuergröße ab. Bei direkter Steuerung müssen Anwendungen ihre Anpassungen in der vom System vorgegebenen Weise formulieren. Dies kann je nach System die Verhaltensweisen beschränken. Bei Ansätzen mit indirekter Steuerung ist dies anders. Die Anpassung muss zwar explizit durch die Anwendung selbst implementiert werden, diese hat je-

Tabelle 5.1: Bewertung der verschiedenen Ansätze zur Steuerung des Energieverbrauchs

	Ladestandsfassung	Verbrauchserfassung	Plattform	Steuerung	Flexibilität	Isolation	Kurzfristig		Langfristig		Integration	Vorbereitung	Laufzeitkosten
							Mehrbedarf	Minderbedarf	Mehrbedarf	Minderbedarf			
Odyssey [191]	/	HW	Linux/PC	Service Level	o	o	+	+	+	+	-	o	o
Inprecise Computations [192]	SW	SW	WSN (Knoten)	Task	-	-	+	o	+	+	-	+	+
Energy Levels [21]	SW	SW <sup>1</sup>	WSN (Knoten)	Service Level	o	-	+	+	o	+	-	-	+
Eon [66]	HW	HW	WSN <sup>2</sup> (Knoten)	Timer/Service Level	o	o	+	+	o	+	o	o	+
IDEA [193]	HW	SW	WSN <sup>2</sup> (Nachbarschaft)	Service Level	o	-	+	+	o	+	-	-	o
EMA [194]	/	HW	WSN (Knoten)	Timer	-	+	o	-	o	+	-	-	-
Variability-aware Software Adaption [195]	/	SW	WSN (Knoten)	Timer	-	o	-	-	-	-	+	-	+
PControl [23]	SW	/	WSN (Knoten)	Timer	-	-	-	-	-	+	+	+	-
Virtual Battery [197]	/	SW	WSN (Knoten)	Energiezuteilung	++	o	++	++	-	-	/	/	+
SORA [198]	/	SW <sup>1</sup>	WSN (Netzwerk)	Preis	o	-	-	-	-	-	+	-	-
Cinder [199]	/	SW	Mobiletelefon	Energiezuteilung	++	++	-	+	-	+	++	o	-
ECOSystem [69]	HW	SW	PC	Energiezuteilung	+	++	-	+	o	o	++	o	+
Pixie[201]	/	SW	WSN (Knoten)	Energiezuteilung	+	+	+	-	-	+	+	o	-
Nemesis[68]	HW	Hybrid	PC	Preis	+	+	-	-	-	-	-	o	o

<sup>1</sup> statisch<sup>2</sup> Harvesting

doch alle Freiheiten, die ihr zugeteilte Energie zu verwalten. Das wirkt sich auch auf den Vorbereitungsaufwand aus. Dieser sollte bei direkter Steuerung grundsätzlich geringer sein, da sich Anwendungen keine Gedanken um die Bedeutung von Energie machen müssen. Der Verbrauch wird vom System überwacht und eingestellt. Im Gegensatz dazu muss bei den Ansätzen mit indirekter Steuerung die Applikation ein Energiebewusstsein besitzen, denn sie entscheidet autonom, was mit der zur Verfügung stehenden Energie passiert. In einem einfachen Fall bedeutet dies beispielsweise das Einstellen des eigenen Intervalls, es kann aber beliebig komplexe Anpassungen umfassen. Betrachtet man die Umsetzung der Konzepte in den Ansätzen, zeigt sich, dass der Vorbereitungsaufwand zum Teil durch das Erstellen von statischen Verbrauchsgrößen (Energy Levels [21], SORA [198]), das Messen des Einflusses der Adaption auf andere Knoten (IDEA [193]) und der Preisgestaltung (SORA [198]) erhöht wird.

Der Vorbereitungsaufwand steigt durch den Aufwand, welcher aus softwaretechnischer Sicht für die Integration der Ansätze in die Anwendung notwendig ist. Ansätze, die Service Level als Steuergröße besitzen, machen eine Anpassung der Anwendung durch Integration dieser Level notwendig (Odyssey [191], Inprecise Computations [192], Energy Levels [21], IDEA [193]). Wird für die Steuerung des Verbrauchs auf die Konfiguration von Timern zurückgegriffen, verringert sich der Aufwand der Integration deutlich, da der Anwendungscode selbst meist nicht bearbeitet werden muss (Variability-aware Software Adaption [195], PControl [23]). Sonderstellungen nehmen Eon [66] und EMA [194] ein. Eon setzt auf eine Kombination aus Service Levels und Timern, wodurch der Aufwand vom gewünschten Anwendungsverhalten abhängt. EMA verwendet zwar Timer zur Steuerung, macht jedoch eine Anpassung der Anwendungsteile durch Annotation notwendig. Ansätze, die den Energieverbrauch indirekt steuern, kommen zum Teil ohne Anpassung der Anwendung aus (Cinder [199], ECOSystem [69]) oder das Energiemanagement ist Teil des Abarbeitungsrahmenwerkes und benötigt keine beziehungsweise kaum Anpassungen der Anwendungskomponenten (Pixie [201], SORA [198]). In Nemesis [68] wird der Energieverbrauch durch den Schattenpreis zwar indirekt gesteuert, jedoch muss sich die Anwendung an eine Veränderung des Preises anpassen und muss somit modifiziert werden.

Die Laufzeitkosten werden von der Häufigkeit bestimmt, in der das System aktiv wird. Erfolgt eine Steuerung kontinuierlich (EMA [194], PControl [23], Pixie [201]) oder wird kontinuierlich Energie transferiert (Cinder [199]), ist der notwendige Aufwand entsprechend hoch. Eine Sonderstellung nimmt SORA ein, da bei diesem Ansatz durch die Preisverbreitung per Funk ein hoher Kommunikationsaufwand notwendig ist. Dies trifft auch auf IDEA zu, da Informationen mit den Nachbarn ausgetauscht werden.

Aufgrund der verschiedenen Anwendungsmöglichkeiten von tief eingebetteten Systemen sind die Anforderungen der Anwendungen an ein Management, wie am Anfang des Kapitels dargestellt, unterschiedlich. Zwar kann Energiemanagement mit direkter Steuerung einen weiten Bereich abdecken, es ist aber nicht zu komplexen Verhaltensanpassungen in der Lage. Indirekte Steuerung durch Ressourcenzuteilung ermöglicht den Anwendungen hingegen sowohl eine umfangreiche Eigensteuerung als auch die Nachbildung von Methoden der direkten Steuerung. Dies setzt sowohl eine Isolation von konkurrierenden Anwendungsteilen als auch die Fähigkeit zur Behandlung von Ausnahmesituationen durch kurz- und langfristige Bedarfsanpassung voraus. Gerade diese durch veränderten Bedarf ausgelöste Dynamik ist bei bisherigen Ansätzen kaum vorhanden.

Im Stand der Technik wurden drei Verfahren zur Zuteilung von Ressourcen vorgestellt. Neben der einmaligen Verteilung kann diese auch periodisch erfolgen oder in Form eines *Token Buckets* kontinuierlich sein. Der notwendige Aufwand ist bei der einmaligen Zuteilung am geringsten und bei der kontinuierlichen Zuteilung am größten. Eine periodische oder kontinuierliche Ausschüttung ermöglicht neben der Limitierung des Verbrauchs auch eine Anpassung an den Bedarf und die Energieverfügbarkeit. Beides ist bei einer einmaligen Verteilung nur schwer möglich, da zum Anpassungszeitpunkt die zugeteilte Energie einiger Anwendungsteile schon verbraucht sein könnte und die Anpassung zu Lasten der sparsameren Teile gehen kann.

Die periodische und die kontinuierliche Zuteilung ähneln sich konzeptionell, unterscheiden sich aber in der Periodendauer und damit dem Planungshorizont der Anwendungen. Während eine Anwendung sich bei periodischer Zuteilung die Energie einteilen kann, ist eine längerfristige Planung bei der kontinuierlichen Ausschüttung nur eingeschränkt möglich, da sich der Zufluss jederzeit verändern kann. Die so entstehende Fokussierung auf den Augenblick ist für bestimmte Anwendungsszenarien jedoch von Vorteil, da bei einem Ereignis das Verhalten anhand der gerade verfügbaren Energie bestimmt werden kann und die Zukunft keine Rolle spielt.

Aufgrund des geringeren Aufwands und der für die Anwendung besseren Möglichkeit der Planung wird in dieser Arbeit ein periodischer Ansatz umgesetzt. Sollte für Anwendungen eine kontinuierliche Energiezuteilung sinnvoll sein, kann dieses Verhalten logisch aus der periodischen Verteilung erzeugt werden. Dies gilt auch für die von direkter Steuerung verwendeten Mechanismen. Das Management sollte in der Lage sein, die Berechnung von Intervallen und Energiestufen auf Grundlage des Verbrauchs für die Anwendung transparent bereitzustellen.

Die Steuerung eines Sensornetzes ist auch durch einen rein lokalen Ansatz möglich. Abbildung 5.3 zeigt ein simuliertes Beispielnetzwerk mit 11 mal 11 Knoten in einer Rasteranordnung. Jeder Knoten hat ein variables Messintervall und muss potenziell Nachrichten der Nachbarknoten weiterleiten. Durch die Verwendung eines

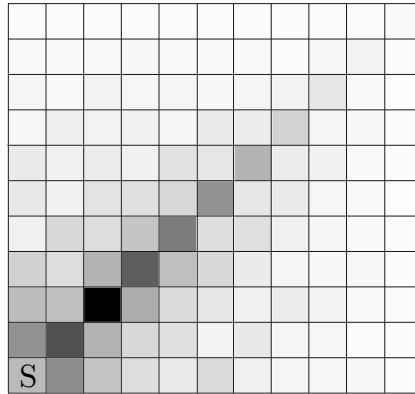


Abbildung 5.3: Unterschiedliche Messintervalle in einem simulierten Sensornetz durch Balancieren der Last mit nicht uniformer Netzwerkauslastung: Das Intervall des Sensors reicht von 1,71s (weiß) bis 3,36s (schwarz).

Routingbaums ist der Kommunikationsaufwand der Knoten nahe der Senke höher. Diese müssen ihr Messintervall anpassen, um den Energieverbrauch zu reduzieren und das Lebenszeitziel zu erreichen. Knoten am Rand profitieren im Gegenzug vom geringeren Aufwand für die Kommunikation. In diesem Beispiel messen die äußeren Knoten alle 1,71 Sekunden, während die inneren Knoten seltener messen und das längste Intervall mit 3,36 Sekunden fast doppelt so lang ist. Durch diese Balancierung können alle Knoten das Lebenszeitziel erreichen, ohne den zusätzlichen Energiebedarf eines globalen Ansatzes. Grundsätzlich kann ein lokaler Ansatz auch als Basis für ein netzwerkweites Energiemanagement dienen.

### 5.3 Dynamische Energiebudgets

Ein Energiemanagement steuert den Verbrauch des Systems, sodass Zielsetzungen für die Lebenszeit erreicht werden. Anwendungen müssen sich an die Vorgaben des Managements hinsichtlich der Entladerate anpassen. Hierzu müssen sie mit Informationen über die verfügbare Energie versorgt werden. Eine Steuerung ist jedoch nur dann möglich, wenn die Energieverfügbarkeit mit einem Zeitfenster assoziiert wird. Beide Informationen können über Energiebudgets bereitgestellt werden. Energiebudgets sind ein abstraktes Reservoir für Energie und repräsentieren das Recht auf eine bestimmte Energiemenge. Sie ermöglichen es Anwendungen, Entscheidungen auf Basis der Energieverfügbarkeit zu treffen und Parameter individuell anzupassen. Das zugrunde liegende Management stellt Anwendungen Strukturen für solche Anpassungen bereit und vereinfacht so den Entwicklungsprozess. Im Vergleich zu existierenden Ansätzen wird durch Regeln des Managements in stärkerem Maße auf die dynamische Natur des Anwendungsgebietes eingegangen. Für eine Anpassung an

die Energieverfügbarkeit und Bedarfsänderungen einzelner Anwendungsteile werden Mechanismen bereitgestellt.

Die hier vorgestellten Budgets basieren auf dem Konzept der *Ressource Container* [24], fokussieren sich aber wie beispielsweise [185] und [196] auf Energie als einzige betrachtete Ressource. Im Gegensatz zu letzteren dienen Budgets nicht der Erfassung von verbrauchter Energie durch Anwendungs- und Systemteile, sondern von verfügbarer Energie. Die dem System zur Verfügung stehende Energie (oder Teile davon) wird zwischen den Budgets aufgeteilt. Da verschiedene Anwendungsteile und -ziele um die Energie konkurrieren, sind Mechanismen notwendig, sodass die Energieanteile voneinander getrennt ausgewiesen werden können. Somit wird eine lokale Sicht auf die verbleibende Energie ermöglicht. Energie, die von Anwendungsteilen durch Nutzung eines Gerätes verbraucht wird, muss einem Budget zugeordnet und abgezogen werden. Da verschiedene Geräte zur gleichen Zeit verwendet werden können, müssen Energiebudgets nebenläufig nutzbar sein. Bei von mehreren Anwendungsteilen genutzten Geräten wie der CPU muss zwischen Budgets gewechselt werden, um unterschiedliche Anwendungsteile separat erfassen zu können. Ist ein Budget aufgebraucht, kann der assoziierte Anwendungsteil nicht mehr ausgeführt werden, bis das Budget wieder aufgefüllt wird. Mithilfe primitiver Operationen zum Hinzufügen und Entfernen von Energie lassen sich komplexe Operationen wie zum Beispiel der Transfer zwischen Budgets umsetzen.

Ein Energiebudget ist definiert durch die momentan gespeicherte Energiemenge  $b$ . Es wird durch eine Anforderung  $B$  zusammen mit einem Gültigkeitsintervall  $[t_{start}, t_{end}]$ , Bedarfsangaben  $min$  und  $max$  und einem Gewicht  $g \geq 1$  vom System beantragt (siehe Formel 5.1).

$$B = (b, [t_{start}, t_{end}], min, max, g) \quad (5.1)$$

Die Dauer des Gültigkeitsintervalls hängt vom Einsatzszenario ab und kann von wenigen Minuten bis zu mehreren Wochen reichen. Die Gültigkeitsintervalle ermöglichen eine Absicherung für angefangene Vorgänge und Schutz gegen Abbrüche durch spontane Änderungen in der Energieverfügbarkeit. Zu Beginn der Intervalle werden die Budgets gleichzeitig aus der dem System verbleibenden Energie aufgefrischt, sodass die Applikation nur intervallweise planen muss.

Ohne Angabe des Bedarfs existiert für das Management keine Verbindung zwischen  $b$  und dem tatsächlichen Bedarf, wodurch die Verteilung der Energie die Anforderungen unzureichend widerspiegeln kann. Die bei anderen Ansätzen umgesetzte Verteilung auf Grundlage von vom Nutzer angegebenen statischen prozentualen Verhältnissen spiegelt oftmals nicht die realen Anforderungen wider. Meist existiert eine Spanne, innerhalb derer der tatsächliche, von äußeren Umständen abhängige Verbrauch liegt. Zusätzlich sollen Anwendungen sich im Sinne des Managements selbst



an die zur Verfügung stehende Energiemenge anpassen. Der zu erwartende Bedarf an das Budget ist somit definiert durch die minimal von den assoziierten Verbrauchern benötigte Energie  $min$  und die maximal verbrauchbare Energie  $max$  während  $[t_{start}, t_{end}]$ . Diese Werte geben dem System sinnvolle Grenzen für das Befüllen von  $b$  zu  $t_{start}$ . Die Bedarfsgrenzen sind abhängig von verschiedenen Faktoren, insbesondere der Anwendung. Die initiale Ermittlung kann sowohl statisch (Experimente, Simulation oder Abschätzung) als auch dynamisch erfolgen. Letzteres erscheint jedoch nur in zyklischen Anwendungen einer Lernphase sinnvoll. Besitzt ein assoziierter Anwendungsteil/Verbraucher keine oder kaum Anpassungsmöglichkeiten, beispielsweise bei der Bündelung des Energieverbrauchs der Schlafzustände des Systems in ein Budget, ist dieser auf die Deckung des gesamten Bedarfs angewiesen. Dies kann grundsätzlich durch entsprechend nahe beieinander liegende oder identische  $min$ - und  $max$ -Angaben erreicht werden.

Die in einem Intervall  $t$  zur Verfügung stehende Energie  $E_{avail}(t)$  lässt sich aus verbleibender Energie  $E(t)$  und verbleibenden Intervallen  $t_{remain} = T - t$  errechnen (siehe Formel 5.2).

$$E_{avail}(t) = \frac{E(t)}{t_{remain}} \quad (5.2)$$

Durch die im Voraus zu ermittelnden Verbrauchsgrenzen lassen sich die Laufzeit und die notwendige Energieversorgung planen. Wie in Formel 5.3 beschrieben, gilt als essenzielle Voraussetzung während des Betriebs, dass der minimale Bedarf für alle Budgets  $b_0 \dots b_n$  in allen verbleibenden Intervallen gedeckt werden kann. Gleichzeitig soll nicht mehr Energie verfügbar sein, als von der Anwendung nutzbar ist. Dabei sollte die Energiemenge initial ausreichen, um einen Spielraum für die Regelung zu schaffen.

$$\forall t < T : \sum_{i=0}^n min_i \leq E_{avail}(t) \leq \sum_{i=0}^n max_i \quad (5.3)$$

Steht nicht mehr genug Energie zur Verfügung, muss zwingend der Nutzer informiert werden, da das Erreichen des Lebenszeitziels und damit die Funktionalität nicht länger garantiert werden können. Die noch verbleibende Energie kann auf verschiedene Arten unter den Budgets verteilt werden, um einen Teilbetrieb sicher zu stellen. Steht dagegen mehr Energie als nutzbar zur Verfügung, hat dies zwar keine Auswirkungen auf das Erreichen des Lebenszeitziels, deutet aber auf eine falsche Planung hin. Eine Information an den Nutzer kann sinnvoll sein, wenn dieser durch etwaige Fernwartungsmechanismen das Maximum der Budgets anpassen kann, um von der überschüssigen Energie zu profitieren.



Für alle Budgets gilt grundsätzlich Formel 5.4. Abweichungen sind je nach verwendeten Regeln möglich. So kann beispielsweise Energie angespart werden, sodass  $max$  temporär überschritten wird.

$$min \leq b \leq max \quad (5.4)$$

### 5.3.1 Zuordnung der Budgets zu Systemkomponenten

Energiebudgets werden mit Systemkomponenten und Verbrauchern in verschiedenen logischen Ebenen assoziiert. Da die Energie durch Hardware verbraucht wird, ist die Bindung an ein Gerät beziehungsweise einen Zustand naheliegend. Durch Modifikation der Treiber ist es zudem möglich, Budgets für gemeinsam genutzte Ressourcen mit deren Anfragen zu verbinden. Dies gilt auch für den Prozessor, wodurch Tasks, Prozesse oder Aktivitäten individuell mit einem Budget versehen und damit individuelle Anwendungsteile energetisch voneinander getrennt werden können. In einem größeren Maßstab werden Energiebudgets auch auf ganze Ausführungspfade einer Anwendung angewendet. Dazu wird in einem ereignisbasierten System einem Ereignis ein Budget zugeordnet. Dieses pfadbasierte Budget wandert dann mit dem Ereignis durch das System und wird für alle von diesem Ereignis ausgelösten Hardwareaktivitäten genutzt. Ein solches Verhalten wurde bereits in Ansätzen zur Verbrauchserfassung ([196] und [187]) demonstriert und beim indirekten Management von Pixie [201] diskutiert. Idealerweise geschieht die Weitergabe des mit dem Ereignis assoziierten Budgets implizit und für den Nutzer transparent.

Abbildung 5.4 stellt den Ereignisfluss der bereits in Abbildung 2.12 auf Seite 24 vorgestellten vereinfachten Sensornetzanwendung dar. Ein Sensor (zum Beispiel ein

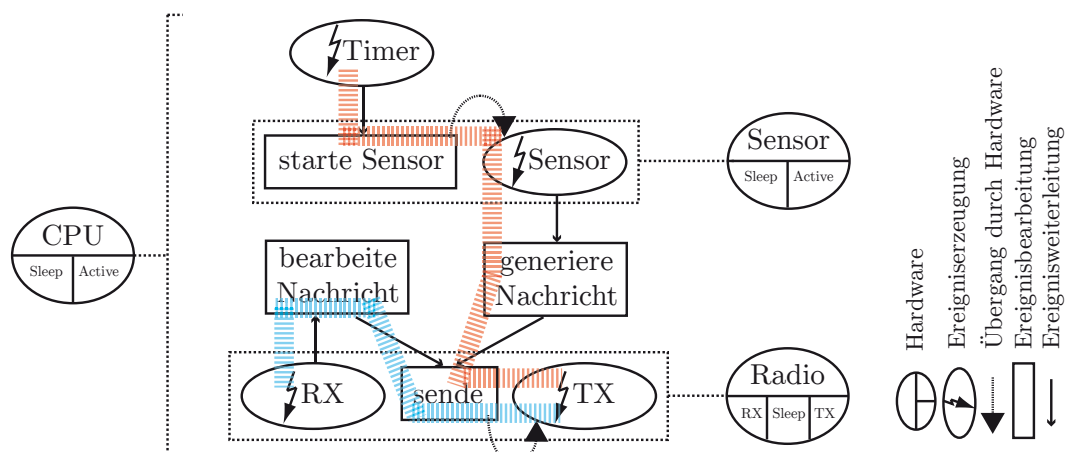


Abbildung 5.4: Ereignisfluss einer vereinfachten Sensornetzanwendung zur Datenerfassung und Weiterleitung mit CPU, einem Sensor und einem Funkmodul: Zusätzlich sind die zwei Ausführungspfade der Anwendung farblich hervorgehoben.

Analog-Digital-Wandler mit mehreren Kanälen) wird von einer zeitgesteuerten Softwareaktivität (starte Sensor) angestoßen. Die gesammelten Daten werden durch eine Aktivität (generiere Nachricht) ausgewertet und verschickt. Zusätzlich können empfangene Nachrichten weitergeleitet werden. Abbildung 2.12 stellt auch die genutzte Hardware dar: Die Software wird von der CPU ausgeführt, Daten vom Sensor gesammelt und Nachrichten durch das Funkmodul empfangen und verschickt.

Gerätebasierte Budgets erfassen nur den gesamten Verbrauch aller Nutzer des Gerätes, eine Trennung ist ohne Weiteres nicht möglich. Für dieses Beispiel bedeutet dies, dass beispielsweise keine Unterscheidung zwischen eigenen und weitergeleiteten Nachrichten möglich ist. Für einfache Anwendungen ist dies auch nicht unbedingt notwendig. Ein einzelnes Budget für das System oder eine Trennung von Sampling und Kommunikation reichen bereits, um die Arbeitsintervalle aufgrund der verbrauchten und verbleibenden Energie einzustellen. Zusätzlich ist eine Assoziation der Budgets zu Gerätezuständen möglich. Auf diese Weise kann beispielsweise Energie, die in aktiven Zuständen verbraucht wird, vom Verbrauch in den Schlafzuständen getrennt verwaltet werden.

Anfragebasierte Budgets realisieren hingegen eine Trennung zwischen einzelnen Aktivitäten und ermöglichen so ein feingranulares Management. Beispielsweise lassen sich so die zeitgesteuerten Softwarekomponenten in Abbildung 5.4 unabhängig voneinander auf Grundlage ihres jeweiligen Verbrauchs steuern. Zu diesem Zweck müssen jedoch alle zusammengehörigen Aktivitäten mit dem jeweiligen Budget assoziiert werden. Die Aktivitäten selbst müssen ebenso modifiziert werden, um Anfragen an den Sensor oder das Radio mit dem korrekten Energiebudget zu versehen.

Dieser Aufwand kann bei pfadbasierten Budgets deutlich gesenkt werden, da dort die Weitergabe der Budgets implizit erfolgt. In der in Abbildung 5.4 dargestellten Anwendung existieren zwei Ausführungspfade (rot und blau). Diese müssen nicht komplett durchlaufen werden. Im Beispiel könnten nur bestimmte Daten versendet oder nicht alle Nachrichten weitergeleitet werden.

Timer→starte Sensor→Sensor→generiere Nachricht→sende→TX  
RX→bearbeite Nachricht→sende→TX

Wird den auslösenden Ereignissen jeweils ein Energiebudget zugewiesen, können alle Pfade mit einer unterschiedlichen Menge Energie ausgestattet werden. Die Steuerung kann dann entweder innerhalb eines Pfades durch eine beteiligte Aktivität erfolgen oder zentral durch eine anwendungsspezifische Softwarekomponente.

Auf welcher Ebene die Budgets letztlich in einer Anwendung verwendet werden, hängt maßgeblich von deren Komplexität und der Zahl nebenläufiger Anwendungsziele ab. Eine Mischung der Ebenen ist ebenso denkbar, insbesondere da bei anfrage- und pfadbasierten Budgets der Verbrauch von Schlafzuständen nicht ohne Weiteres integriert ist, aber mithilfe von gerätebasierten Budgets einbezogen werden kann.

### 5.3.2 Zuordnung von asynchronem Verbrauch

Bei direkt geteilten Ressourcen oder wenn der Verbrauch asynchron erfolgt, muss die in [196] geforderte Fairness gewährleistet werden, sodass nur die Verursacher des Energieverbrauchs belangt werden. Abbildung 5.5 zeigt einen solchen einfachen fairen Aufbau am Beispiel einer SD-Karte. Ein Schreibzugriff der Anwendung schreibt Daten nur in einen Puffer und erst wenn dieser gefüllt ist, wird tatsächlich auf die Karte geschrieben und somit Energie verbraucht. Ohne Gegenmaßnahmen müsste das Budget, das mit dem Hardware-auslösenden Schreibzugriff verbunden ist, mit den gesamten Kosten belastet werden.

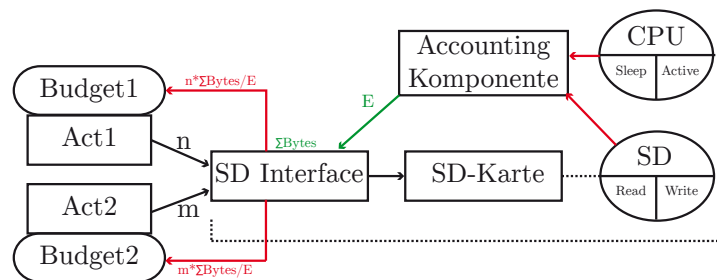


Abbildung 5.5: Beispiel für die faire Abrechnung einer geteilten Ressource: Neben dem Datenfluss (schwarz) sind Energiefluss (rot) und Informationsfluss (grün) dargestellt.

Um dies zu verhindern, muss die Abrechnung der Schreibzugriffe der Anwendung asynchron erfolgen, das heißt Anwendungsteile bezahlen für einen Verbrauch, der erst in der Zukunft erfolgt. Da die eigentlichen Kosten oftmals nicht genau vorhersehbar sind, beispielsweise variiert die Dauer von Schreiboperation von SD-Karten, ist eine Berechnung im Voraus nicht sinnvoll. Im Beispiel in Abbildung 5.5 werden in der Interface-Aktivität die durchschnittlichen Kosten pro Byte aus den tatsächlichen Schreibkosten inklusive Prozessor und den insgesamt geschriebenen Bytes berechnet. Dazu werden die involvierten Geräte und Aktivitäten mit einer speziellen Accounting-Komponente verbunden, die nur den Verbrauch erfasst. Auf diese Weise lassen sich auch Initialisierungs- und Pflegekosten auf alle Nutzer umlegen.

Das Problem der Zuordnung verschärft sich, wenn der Verbrauch von externen asynchronen Ereignissen unterschiedlichen Budgets zugeordnet werden muss. Ein für drahtlose tief eingebettete Systeme relevantes Beispiel ist der MAC. Während die Kosten des Sendens inklusive etwaiger Bestätigungen und Retransmissionen mit der vorgestellten Methode erfasst und umgelegt werden können, kann das asynchrone Empfangen von Nachrichten in den meisten Fällen nicht direkt zugeordnet werden. Dies ist im *idle listening* begründet, dessen Kosten von der Frequenz abhängen, in welcher Nachrichten empfangen werden. Das wiederum ist von der Anwendung und der Umgebungssituation abhängig ist und vom Knoten selbst nicht gesteuert. Wer-

den diese Kosten auf die empfangenen Bytes umgelegt, sind sie höher, wenn selten Nachrichten empfangen werden. Das verkompliziert die Planung für den Zielanwendungsteil. Dies lässt sich umgehen, indem die Kosten des *idle listening* ähnlich wie die Kosten für Schlafzustände als Allgemeinkosten betrachtet und mit einem separaten Budget versehen werden.

## 5.4 Strategien zur Budgetverwaltung

Das Konzept der Energiebudgets lässt eine Vielzahl unterschiedlicher Strategien und Regeln zu. Diese umfassen neben der Verteilung der Energie auf die Budgets ihr Verhalten in Ausnahmesituationen. Zu Letzterem zählen Mehr- und Minderbedarf der mit einem Budget assoziierten Anwendungsteile beziehungsweise Geräte und die allgemeine Energieverfügbarkeit.

### 5.4.1 Verteilung der Energie

Zur Laufzeit muss die zur Verfügung stehende Energie auf die Budgets verteilt werden, sodass das Lebenszeitziel erreicht werden kann. Die Verteilung kann - beispielsweise durch eine prozentuale Aufteilung - durch die Anwendung festgelegt werden oder durch Gewichte eingeschränkt sein. Ohne genaue Vorgaben muss das System selbstständig in der Lage sein, die verfügbare Energie so zu verteilen, dass jedes Budget ohne unbeabsichtigte Bevorzugung so weit wie möglich gefüllt ist. Im weiteren Verlauf der Arbeit wird dies als Fairness von Algorithmen zur Verteilung gefordert.

Algorithmen zur Ressourcenplanung aus dem Betriebssystemkontext wie beispielsweise der Bankieralgorithmus [202] sind für die Verteilung von Energie nicht direkt nutzbar, da sie auf nicht verbrauchbaren Ressourcen basieren. Zwar ist eine Modifikation wie in [203] möglich, jedoch spielt Energie darin nur eine untergeordnete Rolle im Zusammenspiel mit weiteren Ressourcen.

Die Verteilung kann als Ressourcenallokationsproblem interpretiert werden, bei welchem der Nutzen  $U$  aller Ressourcen maximiert werden soll. Hierzu können neben der Brute-Force-Methode Verfahren aus der (ganzzahligen) linearen Optimierung genutzt werden. Das in diesem Fall vorliegende Allokationsproblem lässt sich wie in Formel 5.5 als Optimierungsproblem für  $i$  Budgets darstellen, bei dem maximal  $E$  Einheiten Energie verteilt werden können.

$$\max(\sum U_i(E_i)) \tag{5.5}$$

Hinzu kommen Nebenbedingungen, die einzuhalten sind. So muss jedes Budget ausreichend, aber nicht zu viel Energie erhalten (Formel 5.6), und die verteilte Energie  $E$  muss ausreichen, um alle Budgets zu füllen (Formel 5.7).

$$\min_i \leq E_i \leq \max_i \quad (5.6)$$

$$\sum E_i \leq E \quad (5.7)$$

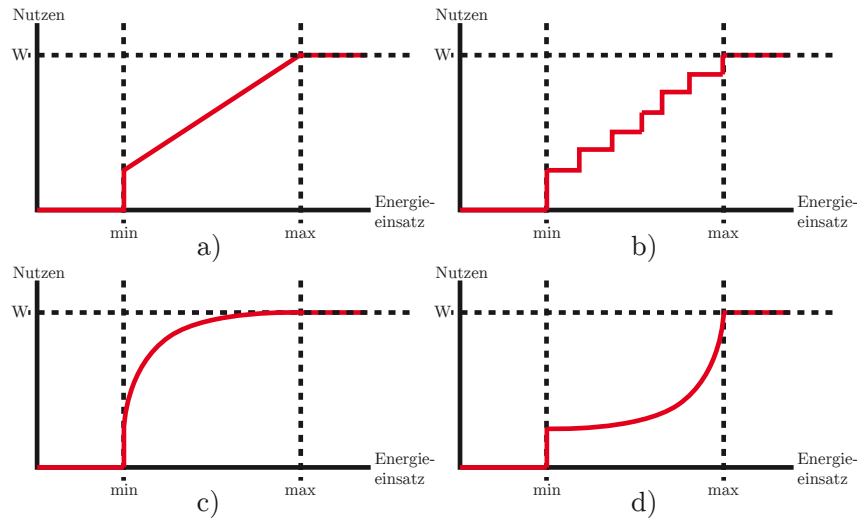


Abbildung 5.6: Typische Nutzenfunktionen für den Energieeinsatz

Die zu diesem Zweck verwendeten Nutzenfunktionen  $U_i$  sind wie  $\min_i$  und  $\max_i$  abhängig von den mit dem Budget assoziierten Anwendungsteilen. Die Funktionen können, wie in Abbildung 5.6 dargestellt, von unterschiedlicher Gestalt sein. Da die Komplexität des Verteilungsproblems im direkten Zusammenhang mit den Nutzenfunktionen steht, kann das Problem durch eine Beschränkung auf a) verringert werden. Die Möglichkeiten werden dadurch nur geringfügig eingeschränkt, da durch Linearisierung (im Falle von b) ) sowie Verschiebung von  $\min$  und  $\max$  (für c) und d) ) aus allen Nutzungsfunktionen eine Funktion der Gestalt von a) zumindest interpoliert werden kann.

Aber auch mit dieser Vereinfachung ist die Komplexität noch zu hoch, um ohne Weiteres auf einem Knoten gelöst werden zu können. Zudem kann die Optimierung auch solche Lösungen finden, die zwar korrekt, jedoch aus Sicht des Gesamtsystems ungünstig sind. So ist es beispielsweise möglich, dass Budgets mit dem höchsten Nutzen die gesamte Energie erhalten. Um dies zu umgehen, müssten weitere Nebenbedingungen hinzugefügt werden, die den Nutzen als Gewichtung für einen fairen Mix aller Budgets verwenden. Dadurch wird das Problem aber weiter verkompliziert. Eine einfache Lösung ist durch eine Heuristik möglich, wobei eine optimale Lösung nicht mehr garantiert werden kann. Vor der Verteilung durch eine Heuristik

muss sichergestellt werden, dass jedes Energiebudget mindestens bis zum jeweiligen *min* gefüllt wird. Für die Verteilung der verbleibenden Energie sind verschiedene Heuristiken möglich. Dabei kann der Anteil eines Budgets sowohl bis zum Erreichen des maximalen Bedarfs limitiert als auch unabhängig von der bereits im Budget gespeicherten Energie sein. Letzteres ermöglicht es einem Budget, nicht verbrauchte Energie über mehrere Intervalle anzusparen.

### Energieverteilung anhand der Budgetanzahl

Eine einfache iterative Variante verteilt die in einem Intervall zur Verfügung stehende Energie  $E$  anhand gleicher Teile. Deren Größe berechnet sich aus der Summe der Gewichte  $G$  aller nicht vollen Budgets. Jedes Budget erhält Anteile in Höhe seines Gewichtes. Dabei werden die Budgets durchlaufen und jedes erhält Anteile in Höhe seines Gewichtes gemäß Formel 5.8, wobei die verfügbare Energie entsprechend um die verteilte Energie verringert wird. Wird der angegebene Maximalwert eines Budgets erreicht, bleibt Energie nach einem Durchlauf übrig, sodass die Verteilung mit angepasstem  $G$  wiederholt werden muss.

$$\forall B_i \in B, b_i = \begin{cases} b_i + \frac{E}{G} * g_i & \text{wenn } b_i + \frac{E}{G} * g_i \leq \max_i \\ \max_i & \text{sonst} \end{cases} \quad (5.8)$$

Diese Heuristik bezieht jedoch nicht den tatsächlichen Bedarf der mit den Budgets assoziierten Anwendungsteile ein. Durch die gleich großen Anteile werden Budgets mit einem kleinen *max* Wert gegenüber solchen mit großem *max* bevorzugt.

### Energieverteilung anhand des Anteils eines Budgets am Gesamtbedarf

Eine andere, beispielsweise in ECOSystem [69] verwendete Heuristik verteilt die Energie auf Grundlage des prozentualen Anteils am Gesamtbedarf aller Budgets. Dieser errechnet sich aus der Summe der maximal von jedem Budget benötigten Energie unter Einbeziehung des Gewichtes (siehe Formel 5.9).

$$\forall B_i \in B, b_i = b_i + \frac{(\max_i - b_i) * g_i}{\sum_{j=0}^B ((\max_j - b_j) * g_j)} * E \quad (5.9)$$

Erreichen Budgets durch die Gewichtung *max*, muss die verbleibende Energie auf die übrigen Budgets verteilt werden.

### Verteilung anhand der verbrauchten Energie

Die bisher vorgestellten Heuristiken verteilen die zur Verfügung stehende Energie nur statisch, jedoch kann sich der Bedarf eines Anwendungsteils im Laufe der Zeit verändern. Anstatt als Zielwert der Verteilung das Maximum *max* eines Budgets

zu nutzen, wird versucht, das Budget mit der verbrauchten und, wenn der Anteil der vorigen Phase nicht ausreichte, angefragten Energie  $r$  zu füllen. Dies erhöht unter Umständen die Dynamik bei der Verteilung und ist sowohl für die Verteilung anhand der Budgetzahl (Formel 5.10) als auch anhand des Anteils eines Budgets am Gesamtbedarf (Formel 5.11) möglich. Da bei diesen Heuristiken potenziell nicht die gesamte verfügbare Energie verteilt wird, muss eine andere Heuristik diesen Rest aufteilen.

$$\forall B_i \in B, b_i = \begin{cases} b_i + \frac{E}{G} * g_i & \text{wenn } b_i + \frac{E}{G} * g_i \leq r_i \\ r_i & \text{sonst} \end{cases} \quad (5.10)$$

$$\forall B_i \in B, b_i = b_i + \frac{(r_i - b_i) * g_i}{\sum_{j=0}^B ((r_j - b_j) * g_j)} * E \quad (5.11)$$

#### 5.4.2 Ausnahmesituationen

Sollte die verbleibende Energie nicht mehr ausreichen oder ein oder mehrere Anwendungsteile mehr oder weniger Energie benötigen, als ihren Budgets zugeteilt wurde, muss eine solche Ausnahmesituation behandelt werden. Die Umsetzung kann auf verschiedene Arten erfolgen, wobei die Auswirkungen auf das Gesamtsystem unterschiedlich ausfallen. Ein Minderbedarf ist für einen einzelnen assoziierten Anwendungsteil/Verbraucher nicht problematisch. Ungenutzte Energie sollte möglichst für andere Anwendungsteile nutzbar gemacht werden, wobei ein limitiertes Ansparen für Anwendungen sinnvoll sein kann. Ein unvorhergesehener Mehrbedarf, beispielsweise durch zusätzliche Nachrichten für Retransmissionen, wirkt sich direkt auf die Arbeitsfähigkeit der Task aus. Um diese zu gewährleisten, muss dem Budget mehr Energie zugeführt werden.

#### Unzureichende Systemenergie

Wenn der minimale Bedarf nicht befriedigt werden kann, muss in jedem Fall der Nutzer informiert werden. Zudem ergeben sich verschiedene Reaktionsmöglichkeiten. Zum einen kann das Minimum trotzdem verteilt werden, solange das System noch über Energie verfügt. Dies hat zur Folge, dass das Lebenszeitziel nicht erreicht werden kann, ermöglicht jedoch einen minimalen Betrieb des Systems. Zum anderen kann die in diesem Intervall verfügbare Energie priorisiert verteilt werden. Auf diese Weise kann ein „Notfallmodus“ implementiert werden, mit dem das Lebenszeitziel potenziell erreicht werden kann. Ist nach dem Decken des Minimums der priorisierten Budgets Restenergie vorhanden, kann diese entweder zusätzlich an diese Budgets ausgeschüttet oder für das nächste Intervall gespart werden.

## Fehlbedarf

Ein Fehlbedarf entsteht, wenn die zugeteilte Energie (die zwischen den in der Anforderung angegebenen Werten *min* und *max* liegt) für den assoziierten Anwendungsteil nicht ausreicht oder nicht aufgebraucht werden kann. Der mit dem Budget assoziierte Verbraucher kann falsch mit der Energie planen oder die tatsächlichen Gegebenheiten können deutlich von den Annahmen zur Ausbringungszeit abweichen.

Erhöhter Bedarf kann durch die kooperativen Mechanismen des Leihens und Stehlens gedeckt und überzählige Energie angespart werden. Weiterhin ist es möglich, die zugeteilte Energiemenge anzupassen. Die Auswirkungen sind je nach Strategie unterschiedlich.

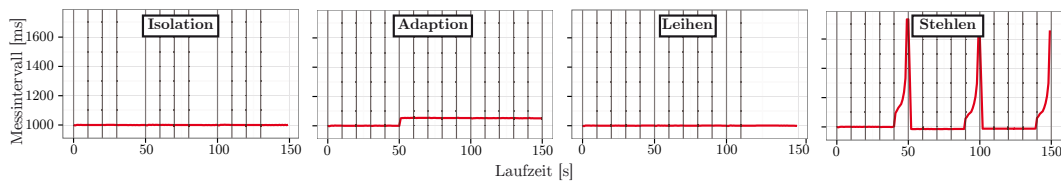


Abbildung 5.7: Einfluss der Strategien, wenn weniger Energie als notwendig zugeteilt wurde

Abbildung 5.7 stellt diese in einem Szenario gegenüber. Bei diesem wurden Experimente auf einem FeuerWhere-Knoten durchgeführt. Die verwendete Anwendung besteht aus zwei Teilen, der Datenerfassung und der Kommunikation, beide mit einem eigenen Budget. Das Intervall der Datenerfassung wurde dynamisch bestimmt und ist in der Abbildung dargestellt. Während der jeweils aus drei Phasen von je 50 Sekunden bestehenden Experimente versuchte der Knoten, fünf Nachrichten zu versenden, hatte initial jedoch nur Energie für die Übertragung von vier Nachrichten pro Phase. In der Abbildung wird diese Kommunikation durch die senkrechten Linien dargestellt.

Bei der Isolation wird die fünfte Nachricht einer Phase nicht übertragen, es gibt aber keinen Einfluss auf die Datenerfassung. Wird der Bedarf des Kommunikationsbudgets adaptiv durch das System angepasst, verändert sich das Intervall der Datenerfassung, da weniger Energie zur Verfügung steht. Kann für die Kommunikation Energie vom System geliehen werden, wird der Energiemangel an das Ende der Laufzeit verschoben. Wird Energie für die Kommunikation vom Datenerfassungsbudget gestohlen, sind die Auswirkungen für dieses, wie dargestellt, enorm. Durch die geringe Restlaufzeit der Phase steigt die Messfrequenz deutlich an, sie fällt nach der Energieverteilung jedoch wieder.

Ein Minderbedarf ist für ein Budget an sich nicht negativ und kann auf unterschiedliche Arten behandelt werden. Kann ein Budget nicht verbraucht werden, stehen die in Abbildung 5.8 exemplarisch dargestellten Möglichkeiten zur Verfügung. Das verwendete Szenario unterscheidet sich nur insofern von dem vorangegangenen,



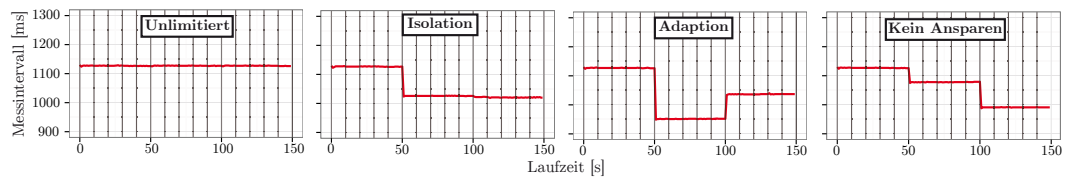


Abbildung 5.8: Einfluss der Strategien, wenn mehr Energie als notwendig zugeteilt wurde

dass hier für die Kommunikation zu viel Energie zur Verfügung steht: Das Budget ist zu groß.

Wird von einem Budget unlimitert Energie aufgenommen, gibt es keine Auswirkungen auf andere Budgets, aber die überschüssige Energie geht logisch gesehen verloren. Besteht eine Beschränkung der Kapazität durch Isolation, kann die überschüssige Energie anderweitig genutzt werden. In diesem Fall sinkt das Messintervall. Gleiches geschieht bei der Anpassung der Kapazität durch die systemseitige Adaption der Budgetgröße. Jedoch sinkt das Messintervall in der zweiten Phase tiefer und stabilisiert sich anschließend. Dies ist darin begründet, dass die überzählige Energie nicht aus dem Kommunikationsbudget entfernt wurde und somit für die Deckung des Bedarfs weniger Energie aufgebracht werden muss. Anschließend ist das Budget vollständig geleert und die zur Füllung benötigte Energiemenge somit etwas größer. Wird überzählige Energie vor der Verteilung aus dem Budget entfernt und somit das Ansparen insgesamt verhindert, steht dem System bei der Verteilung mehr Energie zur Verfügung. Zwar muss das Kommunikationsbudget wieder gefüllt werden, es bleibt aber mehr Energie für die Datenerfassung, wodurch das Messintervall langsam sinkt.

Für jedes Budget kann die Strategie individuell gewählt werden, sodass in einem System beliebige Mischungen existieren können. Die Wahl hängt von der Anwendung und dem vom Nutzer gewünschten Verhalten ab:

**Adaption** eignet sich für alle die Anwendungsteile, die einen nicht genau planbaren Verbrauch haben, aber für das Anwendungsziel essentiell sind. Dazu zählen passive Anwendungsteile wie die Schlafmodi der Geräte oder periodische Servicefunktionalitäten wie die Behandlung von Überläufen der Timer. Auch die Laufzeit und der Verbrauch der Energieverteilung sind von der Zahl der Budgets und der zur Verfügung stehenden Energie anhängig.

**Leihen** eignet sich für alle Anwendungsteile, die einen kurzfristigen, unplanbaren Mehrbedarf erzeugen können. Dies trifft beispielsweise auf adaptives Sampling zu. Die Anwendung kann das gesamte Budget verwenden, ohne auf mögliche zukünftige Ereignisse Rücksicht nehmen zu müssen. Ein Mehrbedarf durch

hohe Intervalle wird durch einen Kredit gedeckt, dessen Tilgung das Intervall in ereignisarmen Phasen senkt.

**Stehlen** bevorzugt Anwendungsteile, deren Funktion wichtiger ist als die anderer Teile. Es kann innerhalb einer Anwendung genutzt werden, um sporadische Verbraucher mit Energie zu versorgen, ohne dass Energie „zwischengespeichert“ werden muss. Es steht der Anwendung also die gesamte Energie zur Verfügung. Anders als bei der Adaption wird jedoch keine Energie der verbleibenden Systemenergie entnommen, sondern bereits im aktuellen Intervall verplante Energie genutzt. Es sind also nicht alle Budgets betroffen, sondern nur ein oder mehrere vorher festgelegte Budgets.

**Ansparen** kann in Anwendungsszenarien sinnvoll sein, in denen sporadische Ereignisse stoßweise auftreten können oder die Anwendung ihren Bedarf aktiv durch adaptives Sampling einstellt. Die Fähigkeit, Energie „zur Seite zu legen“, ermöglicht im letzteren Fall eine größere Entscheidungsfreiheit bei der Wahl der Erfassungsrate. Dem System wird ein Maximum mitgeteilt, das eine mittlere Erfassungsrate ermöglicht und unter dem tatsächlichen Maximalbedarf liegt. So erhält das Budget per se einen geringeren Anteil und nur dann zusätzliche Energie, wenn andere bereits vollständig gefüllt sind.

### Bedarfsdeckung durch Adaption

Eine Adaption passt die angegebenen Werte von *min* und *max* so an, dass das Budget den Bedarf des assoziierten Anwendungsteils befriedigen kann. Dies sollte aus Gründen der Konsistenz und Sicherheit grundsätzlich nicht von der Anwendung selbst aus möglich sein, sondern immer vom Energiemanagement oder dem Nutzer gesteuert werden. Reicht die vorhandene Energie nicht aus, wird der Fehlbetrag (begrenzt) durch das System ausgeglichen und die Grenzen werden im nächsten Intervall entsprechend angepasst. Wird weniger Energie als ursprünglich angenommen verbraucht, wird die überschüssige Energie entfernt und die Grenzen werden angepasst. Ein Ansparen ist für diese Anwendungsteile damit ausgeschlossen.

### Bedarfsdeckung durch Leihen

Leihen bedeutet im Zusammenhang mit Energiebudgets die Erhöhung der Energieverfügbarkeit eines Budgets durch Bereitstellung von zusätzlicher Energie durch das System oder ein anderes Budget, wobei dies als Kredit zurückgezahlt werden muss. Ein solches Verfahren wird von verschiedenen Ansätzen (Virtual Batterie [197], Pixie [201]) angewendet, um Bedarfsspitzen zu decken. Das Zurückzahlen von geliehener Energie wird auf alle folgenden Intervalle aufgeteilt und erfolgt nach der Energie-

verteilung. Die zurückzuzahlende Energiemenge  $z$  berechnet sich, wie in Formel 5.12 dargestellt, aus den verbleibenden Schulden  $s$  und der Restlaufzeit:

$$z = \frac{s_i}{t_{\text{remain}}} \quad (5.12)$$

Da der Kredit zurückgezahlt werden muss, ist die so erwerbbarer Energiemenge begrenzt. Dies kann entweder durch einen statischen Wert umgesetzt oder dynamisch auf Grundlage der verbleibenden Laufzeit und der Rückzahlungsfähigkeit des Budgets berechnet werden. Letzteres stellt sicher, dass die zugewiesene Energiemenge auch nach Abzug der Kreditrate noch über dem minimalen Bedarf liegt, da andernfalls die erneute Kreditaufnahme unausweichlich ist. Im letzten Intervall ist kein Leihen mehr möglich, da dem System keine weitere Energie zur Verfügung steht.

Systemseitig kann das Verleihen auf zwei verschiedene Arten geregelt werden. Die verliehene Energiemenge kann entweder von der Gesamtenergie abgezogen oder nebenläufig in einer eigenen Variable gespeichert werden. Während ersteres sich auf alle Budgets auswirkt, da für alle weniger Energie zur Verfügung steht, die nur langsam wieder zurückgeführt wird, gibt es keine direkten Auswirkungen bei der zweiten Variante. Jedoch muss dort bei der Betrachtung der verbleibenden Energie die verliehene und damit verbrauchte Energie einbezogen werden.

### Bedarfsdeckung durch Stehlen

Stehlen ist im Zusammenhang mit Energiebudgets die Erhöhung eines Budgets durch Entnahme aus einem anderen Budget, ohne dass eine direkte Kompensation erfolgt. Dies ist grundsätzlich für das bestohlene Budget problematisch, da sich die Verfügbarkeit ändert und eine Anpassung des Verbrauchs erfolgen muss. Ohne Limitierung der entnommenen Energiemenge kann der Betrieb der Verbraucher, denen das Budget zugeordnet wurde, nicht mehr garantiert werden. Daher sollte zum einen das bestohlene Budget informiert werden, sodass der Verbrauch angepasst werden kann, und zum anderen der Betrag limitiert sein. Die Limitierung muss derart gestaltet sein, dass das betroffene Budget noch groß genug ist, um den minimalen Betrieb der angeschlossenen Verbraucher garantieren zu können. Zu diesem Zweck kann unter Annahme eines über die Laufzeit gesehen linearen Verbrauchs für ein Budget der zeitabhängige Minimalbedarf  $\widehat{min}$  innerhalb eines Intervalls der Länge  $D$  und der abgelaufenen Zeit  $d$  berechnet werden (Formel 5.13).

$$\widehat{min} = min * \frac{d}{D} \quad (5.13)$$

Das bestohlene Budget kann entweder statisch festgelegt oder mithilfe des Systems ausgewählt werden. Das System kann das Spenderbudget aufgrund der vorhandenen Energiemenge auswählen oder die Belastung auf mehrere Budgets aufteilen,

die anhand ihres Gewichtes ausgewählt werden. Grundsätzlich bieten sich alle diejenigen Budgets an, deren Energieinhalt über dem zeitabhängigen Maximalbedarf  $\widehat{max}$  liegt (Formel 5.14).

$$\widehat{max} = max * \frac{d}{D} \quad (5.14)$$

Die überschüssige Energie kann dann ohne Einschränkung entzogen werden. Desweiteren sind Kombinationen der vom System ausgesuchten Spenderbudgets möglich: Zuerst wird versucht, den Bedarf aus Budgets mit überschüssiger Energie zu decken. Reicht diese nicht aus, wird Energie gestohlen, beginnend mit den Budgets mit dem niedrigsten Gewicht. Die entnommene Energiemenge ist für jedes Budget durch ihren zeitabhängigen Minimalbedarf begrenzt. Als eine weitere Stufe kann vom System Energie geliehen werden, wenn die gestohlene Energie nicht ausreicht.

### Ansparen von Energie

Ein Minderbedarf ist für ein Budget an sich nicht negativ, hat aber potenzielle Auswirkungen auf das Gesamtsystem. Kann nicht verbrauchte Energie dauerhaft in einem Budget akkumuliert werden, fällt der Ladestand der Batterie aus Sicht des Systems höher als erwartet aus. Wie bei ECOSystem [69] wird durch das Management die verfügbare Energie erhöht. Wird die angesparte Energie dann doch verbraucht, entlädt sich die Batterie schneller als erwartet. Das kann zu einem verstärkten Peukert-Effekt (siehe Kapitel 2.1.1) führen, der die tatsächlich nutzbare Kapazität der Batterie verringert.

Ein Ansparen tritt dann auf, wenn bei der Verteilung die bereits angesammelte Energie nicht einbezogen wird. Existiert überdies bei der Energieverteilung keine Begrenzung, kann Energie unbegrenzt gehortet und damit anderen Budgets vorenthalten werden. Ist ein Ansparen von der Anwendung gewünscht, ist deshalb ein Limitierungsmechanismus über das Maximum des Budgets hinaus notwendig.

Die Limitierung über ein vom Bedarf abhängiges Maximum wurde in ECOSystem [69] umgesetzt. Da mit den Budgets bereits das  $max$  zur unmittelbaren Begrenzung existiert, muss ein weiterer Mechanismus hinzugefügt werden, der ein begrenztes Ansparen erlaubt. Dieser kann entweder durch einen statischen Wert repräsentiert sein oder sich adaptiv mit der Laufzeit verändern. Das dynamische Limit  $C$  (siehe Formel 5.15) repräsentiert einen prozentualen Anteil  $\kappa$ , der von den mit dem Budget assoziierten Verbrauchern über die verbleibende Laufzeit konsumierbaren Energie. So ist das Limit zum Anfang der Laufzeit entsprechend hoch, sinkt aber mit Voranschreiten der Zeit kontinuierlich und erreicht im letzten Intervall 0.

$$C = \kappa * (max * (t_{remain} - 1)) \quad (5.15)$$

## 5.5 Verwendung von Budgets zur Kontrolle der Anwendung

Der beschriebene Ansatz der Energiebudgets hat neben der Verteilung von Energie das Ziel, Anwendungen im Bedarfsfall Anreize zur Verhaltensänderung zu geben. Die Anwendung selbst muss auf diese Anreize reagieren, beispielsweise durch Adaption ihres Intervalls. Andernfalls könnten dank der Isolationsfunktion der Budgets die Geräteaktivierungen durch die Treiber verhindert und eine durchgehende Funktionalität nicht sichergestellt werden.

Wie eine Anwendung letztlich aufgeteilt wird und welchen Teilen Energiebudgets zugeteilt werden, hängt dabei von der Anwendung selbst und der Zahl der zu beeinflussenden „Verbrauchsregler“ ab. Während es möglich ist, für eine Anwendung nur ein Budget für Datenerfassung, -verarbeitung und Kommunikation, zu nutzen, ermöglicht die Nutzung mehrerer Budgets eine feingranularere Steuerung. Durch die Festlegung von eigenständigen Budgets für Datenerfassung und Kommunikation kann beispielsweise einfacher das Messintervall bestimmt werden, da Verbräuche des Funkmoduls keinen störenden Einfluss auf das Budget haben und das ermittelte Intervall homogener wird.

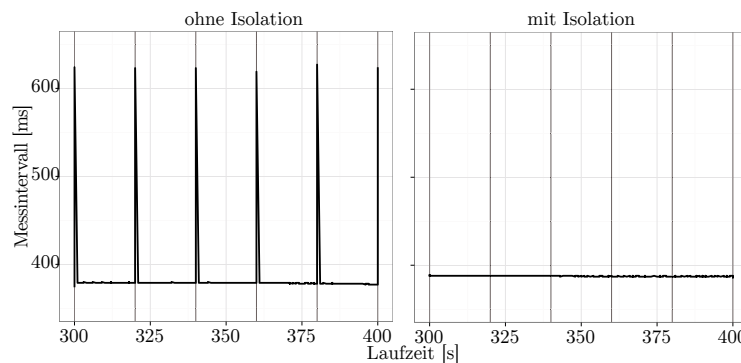


Abbildung 5.9: Dynamische Berechnung des Messintervalls mit und ohne Isolation zwischen Messwerterfassung und periodischer Datenübertragung alle 20s (dargestellt durch vertikale Linien)

Dies ist beispielhaft in Abbildung 5.9 dargestellt. In einem Experiment wurden periodisch Funknachrichten versendet und gleichzeitig Messwerte erfasst. Das Intervall der Messwerterfassung wurde anhand der zur Verfügung stehenden Energie und des Energieverbrauchs innerhalb eines Messintervalls vom System dynamisch bestimmt (siehe Kapitel 5.5.1). Ohne eine Trennung dieser beiden Aufgaben wird das Messintervall deutlich vom Sendevorgang gestört, da keine Unterscheidung zwischen dem Verbrauch der Datenübertragung und des Messvorgangs erfolgt. Werden beide Aufgaben voneinander isoliert, schwankt das Messintervall nur geringfügig.

Energiebudgets stellen Informationen sowohl über die Energieverfügbarkeit als auch den zugehörigen zeitlichen Kontext bereit. Wie die Anwendung diese Informationen nutzt, hängt von den zu beeinflussenden Reglern ab. Bei den in Kapitel 5.2 vorgestellten Ansätzen werden bei der direkten Steuerung zum einen die Anpassung von Intervallen (innerhalb eines tolerierten Bereichs) und zum anderen Servicestufen genutzt. Während Ersteres sich insbesondere für die Datenerfassung durch Sensorik eignet, ermöglicht Letzteres komplexere Veränderungen des Anwendungsverhaltens. Andere Ansätze nutzen ein *Token Bucket*, um die Anwendung zu limitieren. Im Folgenden wird gezeigt, wie sich diese Ansätze mithilfe von Budgets umsetzen lassen.

### 5.5.1 Bestimmung von Anwendungsintervallen

Die Berechnung eines Intervalls  $\tau$  im Sinne des Ausführungsabstandes erfolgt auf Grundlage der verbleibenden Energie, des Zeithorizonts aus der vergangenen Zeit, der Dauer des Intervalls und des Energieverbrauchs der letzten Ausführung  $c$ . Letzteres wird jeweils bei der Berechnung des Intervalls durch Vergleich des Budgets bei der letzten Berechnung ermittelt. Wird nicht bei jeder Ausführung ein neues Intervall berechnet, muss zusätzlich die Zahl der Ausführungen  $n > 0$  erfasst und einbezogen werden (siehe Formel 5.16).

$$\tau = \lfloor \frac{(D - d) * \frac{c}{n}}{b} \rfloor \quad (5.16)$$

Das so berechnete Intervall ist jedoch nicht zwangsweise stabil, sondern wird durch Schwankungen des Verbrauchs und notwendige Rundungen beeinflusst. Da durch Rundungen das Intervall länger ist als das ideale Intervall, wird die zur Verfügung stehende Energie nicht vollständig aufgebraucht und es kann zu Schwankungen zwischen den Intervallen des Energiemanagements kommen.

### 5.5.2 Bestimmung von Servicestufen

Servicestufen beziehungsweise Anwendungsstufen dienen zur Einstellung des Verbrauchs durch das Anpassen des Anwendungsverhaltens. Die Stufen sind geordnet und es wird vorausgesetzt, dass eine größere Stufe einen höheren Energiebedarf hat. Ein Budget, das nur bis zu *min* gefüllt wird, führt zur niedrigsten Servicestufe 1, während *max* zur höchsten Stufe  $N$  führt. Die tatsächliche Servicestufe  $\sigma$  basiert auf dem Unterschied zwischen der verfügbaren Energie und dem zeitabhängigen maximalen/minimalen Energiebedarf (siehe Formel 5.17).

$$\sigma = \begin{cases} 1 & \text{wenn } b \leq \widehat{min} \\ N & \text{wenn } b \geq \widehat{max} \\ \frac{N * (b - \widehat{min})}{\widehat{max} - \widehat{min}} & \text{sonst} \end{cases} \quad (5.17)$$

Die berechneten Stufen haben einen gleichmäßigen Abstand. Sind von Seiten der Anwendung ungleichmäßige Abstände zwischen den Stufen gewünscht, kann diese berechnete Stufen auslassen und ändert die Stufe nur beim Überschreiten von Schwellwerten. Der Verbrauch der jeweiligen Stufen ist bei diesem Vorgehen unbekannt und kann sich durch externe Einflüsse verändern. Daher sollten die Servicestufen periodisch neu berechnet werden.

Da die Servicestufen auf einzelnen Budgets basieren, sind im Gegensatz zu in anderen Ansätzen verwendeten globalen Stufen auch lokale Stufen für einzelne unabhängige Anwendungsteile möglich. Ein typisches Anwendungsszenario für Servicestufen findet sich beim Medienzugriff und der Wegewahl. So kann beispielsweise die Sendestärke in Abhängigkeit von der Stufe gesteuert werden. Außerdem kann die Beteiligung bei der Nachrichtenweiterleitung oder die Zahl von aktiven Slots von den Stufen abhängen.

### 5.5.3 Quantisierung durch Token Buckets

Das beispielsweise in SORA [198] verwendete Konzept der *Token Buckets* dient der Limitierung des Energieverbrauchs. Anwendungen können nicht unmittelbar auf ein Energiebudget zugreifen, sondern nur die aktuell im Bucket angesammelte Energie nutzen. Die Rate  $\gamma$ , mit der ein Bucket in jeder Zeiteinheit befüllt wird, berechnet sich wie in Formel 5.18 dargestellt und hängt von dem verfügbaren Budget und der verbleibenden Zeit ab.

$$\gamma = \frac{b}{D - d} \quad (5.18)$$

Die kontinuierliche Neuberechnung des Bucket-Inhalts stellt einen erheblichen Aufwand dar. Es bietet sich an, dies nur zu tun, wenn ein Zugriff auf das Bucket erfolgt ist oder sich das zugrunde liegenden Budget von außen verändert hat. Die Berechnung des aktuellen Inhalts  $tb$  gemäß Formel 5.19 basiert auf der seit dem letzten Füllen des Buckets vergangen Zeit  $t_{fill}$  und der Rate  $\gamma$ .

$$tb = tb + \gamma * (d - t_{fill}) \quad (5.19)$$

Da eine Anwendung das *Token Bucket* abfragen muss, wenn die aktuelle Menge nicht ausreichend ist, bietet sich ein Dienst an, der die Anwendung bei Erreichen einer bestimmten Energiemenge  $E_{soll}$  benachrichtigt. Dies kann beispielsweise über einen Timer erfolgen, dessen Auslesezeitpunkt durch  $\lceil \frac{E_{soll}}{\gamma} \rceil$  bestimmt wird.

Mithilfe eines Buckets sind Anwendungen in der Lage, nur auf Grundlage der im Bucket verfügbaren Energie zu reagieren und so beispielsweise zwischen einfachen oder komplexen Operationen zu wählen. Eine Vorstellung von Zeit ist für die Anwendung somit nicht nötig.



## 5.6 Evaluation

Für die Evaluation wurde der Ansatz der Energiebudgets für das Betriebssystem REFLEX implementiert. Er baut auf dem in Kapitel 4 vorgestellten Ansatz zur Verbrauchserfassung auf.

Für den Nachweis der Wirksamkeit des Energiemanagements und die Evaluation der vorgestellten Konzepte wurden mathematische Modelle, Simulationen und Experimente mit verschiedenen Szenarien für drahtlose tief eingebettete Systeme durchgeführt. Die Simulationen erfolgten mit der zeitdiskreten Ereignissimulationsumgebung Omnet++ [177]. In diese Umgebung wurde REFLEX integriert [204], es führt die Systemkomponenten des Energiemanagements aus. REFLEX-Komponenten wurden erstellt, die Energie aus den Budgets abziehen und so Verbraucher nachbilden.

In diesem Kapitel wird zunächst der durch das Energiemanagement verursachte Mehraufwand betrachtet. Anschließend werden die Heuristiken zur Verteilung der Energie auf das Erreichen der geforderten Fairness untersucht. Darauf folgt eine Betrachtung der Strategien für den Umgang mit Ausnahmesituationen.

### 5.6.1 Mehraufwand

Das Energiemanagement erzeugt einen Mehraufwand sowohl hinsichtlich des benötigten Speichers als auch hinsichtlich der Ausführungszeit und damit der Energie.

Der Mehrbedarf an Speicher wird in Tabelle 5.2 für die Komponenten und Optionen des Energiemanagements aufgeschlüsselt. Die Treiber müssen zusätzlich für anfragebasierte und pfadbasierte Budgets angepasst werden, sodass Anfragen und Ereignissen das richtige Budget zugeordnet wird.

Für das Management werden im ROM mindestens 3,6kB Speicher benötigt. Hinzu kommt der Bedarf der notwendigen Verbrauchserfassung. Der Speicherbedarf lässt sich durch Verwendung von 32Bit statt 64Bit Integer für die Budgets um rund 10% verringern.

Der Laufzeitaufwand des Energiemanagements hängt maßgeblich von der Ebene ab, auf der Systemkomponenten mit den Budgets assoziiert werden. Da je nach Ebene nicht nur physikalische Zustandsänderungen der Hardware, sondern auch logische, von der Software bestimmte Zustandsänderungen erfasst und mit den Budgets abgerechnet werden müssen, vergrößert sich der Mehraufwand. Um diesen zu ermitteln, wurde die bereits für die Ermittlung des Mehraufwands der Verbrauchserfassung genutzte Anwendung (siehe Kapitel 4.4.1, Seite 91) um das Energiemanagement erweitert. Zwei FeuerWhere-Knoten [189] senden sich im Ping-Pong-Verfahren Nachrichten. Als Maß für den Mehraufwand wird der Einfluss der verschiedenen Managementvarianten auf die Datenrate genutzt. Dazu wurden verschiedene mit



Tabelle 5.2: Speicherbedarf für einzelne Komponenten des Energiemanagements für 64Bit Budgets auf der MSP430 Plattform und den gcc Compiler (Listengröße  $n$ )

	Speicherbedarf in Byte	
	ROM	RAM
Basiskomponenten		
energyBudgetManager <sup>1</sup>	2185	162+2*n
Verteilung Minimalbedarf	316	0
Verteilung gleiche Anteile	376	0
Verteilung Anteil am Gesamtbedarf	626	0
Verteilung tatsächlicher Bedarf	530	0
Budgets		
Energiebudget	1096	80 pro Budget
leihendes Energiebudget <sup>2</sup>	+1368	+8 pro Budget
stehendes Energiebudget	+294	+2 pro Budget
Zurodnungsebene		
gerätebasierte Budgets	0	2 pro Energiesenke
anfragebasierte Budgets	12 <sup>3</sup>	+2 pro Aktivität und Interrupt
pfadbasierte Budgets <sup>4</sup>	+52 <sup>3</sup>	+1+n pro Aktivität

<sup>1</sup> ohne Verteilung

<sup>2</sup> mit Speicherbedarf im Manager

<sup>3</sup> zusätzlicher Aufwand im Treiber

<sup>4</sup> pfadbasierte Budgets enthalten anfragebasierte, diese wiederum gerätebasierte Budgets

Budgets assoziierte Komponenten und unterschiedliche Frequenzen des Prozessors verwendet. Faktoren wie die Verteilung der Energie oder spezielle Situationen und Strategien werden nicht betrachtet, da insbesondere die Verteilung vom Ansatz her nur sehr selten durchgeführt wird.

Die Variante der gerätebasierten Budgets nutzt nur ein globales Budget, das allen Energiesenken im System zugeordnet ist. Bei den anfragebasierten Budgets wird ein zusätzliches Budget genutzt, welches der Sende-anfrage zusammen mit den Daten zugeordnet wird. Die pfadbasierte Variante nutzt ebenso zwei Budgets. Während das erste wie bei den anderen Varianten allen Senken standardmässig zugeordnet wird, ist das zweite Budget mit dem Empfangsereignis des Funkmoduls verbunden und wird von dort aus entlang des Ausführungspfades weitergereicht.

Tabelle B.4 im Anhang zeigt die vollständigen Ergebnisse des Experimentes. Die prozentuale Verringerung der Pakete pro Sekunde und der Mehraufwand gegenüber einer reinen Verbrauchserfassung sind in Abbildung 5.10 sowohl bei der Verwendung von gerätebasierten als auch bei anfragebasierten und pfadbasierten Budgets dargestellt. Analog zur reinen Verbrauchserfassung wirkt sich die geringere Paketrate bei

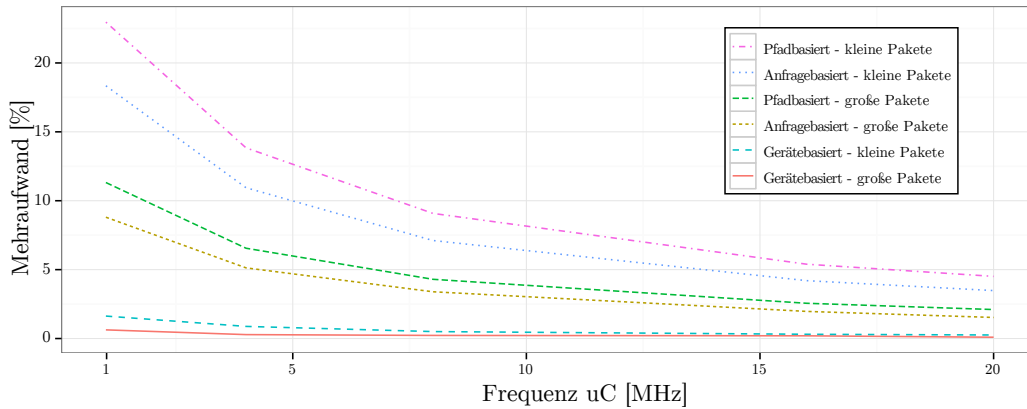


Abbildung 5.10: Mehraufwand des Energiemanagements mit einer Funkanwendung bei zwei Paketgrößen und verschiedenen Frequenzen des Mikrocontrollers: Budgets assoziiert zu Geräten, Anfragen oder Ausführungspfaden

größeren Paketen positiv auf den Mehraufwand aus, da durch längere Wartezeiten bei der Übertragung absolut gesehen weniger Zustandswechsel erfolgen. Deutlich ist der Einfluss der Ebene zu erkennen, auf der die Budgets Verbrauchern zugeordnet werden. Bei gerätebasierten Budgets ist der Mehraufwand vernachlässigbar gering, da zusätzlich zur Verbrauchserfassung nur die erfasste Energie aus den mit den Geräten assoziierten Budgets abgezogen werden muss.

Anders sieht dies bei anfragebasierten Budgets aus. Die notwendigen logischen Zustandswechsel beim Scheduling und der Abarbeitung der Pakete haben einen deutlichen Einfluss auf die Paketrage. Bei jeder Aktivitätsausführung muss im Scheduler das Budget der Aktivität überprüft und potenziell das aktive Budget gewechselt werden. Dies macht eine Abrechnung des Verbrauchs des Prozessors durch die Verbrauchserfassung notwendig. Ein ähnlicher Aufwand entsteht bei den Peripheriegeräten, da zum Teil zusätzliche Abrechnungen für den Budgetwechsel eingefügt werden müssen.

Noch mehr Aufwand verursachen die pfadbasierten Budgets. Dieser entsteht durch die notwendigen Anpassungen des Scheduling, um das Budget des Pfades weiterzuleiten. Bei den Geräten ist der Aufwand ähnlich dem bei anfragebasierten Budgets, nur dass in diesem Fall das Pfadbudget dem Sendevorgang zugeordnet wird.

Das verwendete Szenario stellt wie schon bei der Verbrauchserfassung einen Extremfall dar. Der hohe Mehraufwand bei niedrigen Frequenzen des Prozessors ist bei den üblicherweise genutzten höheren Frequenzen deutlich geringer. Die längeren Inaktivitätsphasen bei den großen Nachrichten halbieren den prozentualen Mehraufwand. In realen Anwendungen mit hohen Standzeiten finden nur wenige Aktivitäten statt, unterbrochen von langen Phasen der Inaktivität. Der Mehraufwand durch die

entsprechend geringe Zahl an physikalischen und logischen Zustandsübergängen ist zwar vorhanden, kann aber meistens vernachlässigt werden.

Neben dem Mehraufwand bei der Ausführung der Anwendung entstehen bei der Energieverteilung weitere Kosten. Diese hängen von der verwendeten Heuristik und der Zahl der Budgets ab. Abbildung 5.11 stellt die experimentell ermittelte Laufzeit der Heuristiken einander gegenüber.

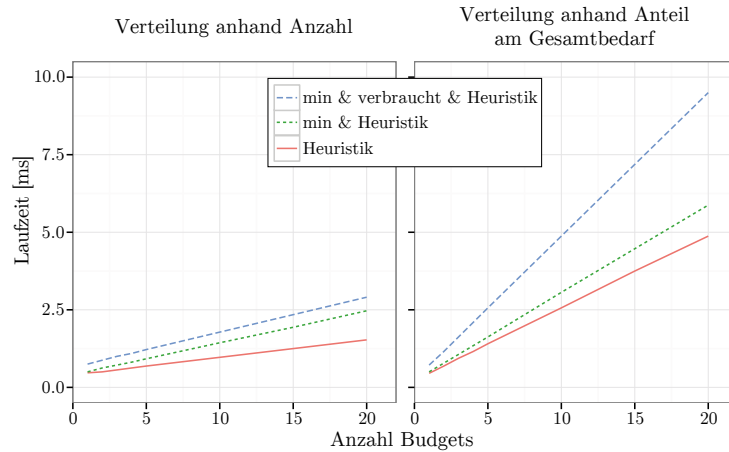


Abbildung 5.11: Laufzeit der Energieverteilung auf MSP430 bei 16MHz: nur die jeweilige Heuristik, mit der zusätzlichen Verteilung des minimalen Bedarfs und zusätzlich mit dem Zwischenschritt durch Verteilung der verbrauchten Energie

Die Laufzeiten der Heuristiken steigen linear mit der Anzahl der verwendeten Budgets. Die Deckung des minimalen Bedarfs und ein Zwischenschritt, der die verbrauchte Energie verteilt, steigern den Laufzeitaufwand entsprechend. Die Verteilung anhand der Budgetzahl benötigt deutlich weniger Laufzeit, was in der geringeren Komplexität begründet liegt. Der Laufzeitaufwand wird jedoch dadurch relativiert, dass die Verteilung nur selten erfolgt. Ein Intervall hat typischerweise eine Dauer von mehreren Minuten bis mehreren Stunden. Bei Anwendungen, die nur einen sehr geringen Energiebedarf haben, kann die Dauer auch noch länger sein.

### 5.6.2 Verteilung der Energie

Die in Kapitel 5.4.1 vorgestellten Heuristiken zur Verteilung wurden mithilfe mathematischer Modelle untersucht. Die in Tabelle 5.3 dargestellten Parameter der drei Budgetanforderungen repräsentierten unterschiedliche Anwendungsanforderungen. Sie wurden so dimensioniert, dass das unterschiedliche Verhalten der Verteilungsheuristiken erkennbar wird.

In Abbildung 5.12 wird die Verteilung der Energie bei der Verwendung der vorgestellten Heuristiken dargestellt. Wie in Kapitel 5.4.1 beschrieben, wurde der mini-

Tabelle 5.3: Verwendete Budgetanforderungen

Energiebedarf pro Intervall	
Budget A min/max	0,9/1,0mJ
Budget B min/max	0,5/3,0 mJ
Budget C min/max	0,6/1,8 mJ

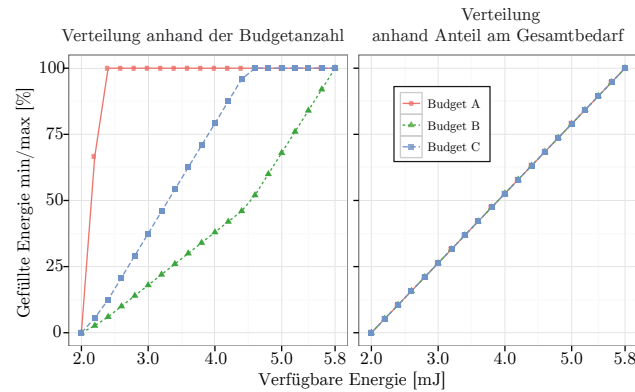


Abbildung 5.12: Energieverteilung zuerst bis  $\min$ : verbleibende Energie anhand der Budgetanzahl und des Anteils eines Budgets am Gesamtbedarf - Verfügbare Energie von  $\sum \min$  bis  $\sum \max$  - Budgetgröße zwischen  $\min$  und  $\max$  in Prozent

male Bedarf vor der Verteilung durch die Heuristik befriedigt. Alle Budgets besitzen dieselbe Priorität.

Erfolgt die Energieverteilung anhand der Budgetzahl, hängt die den Budgets zugewiesene Energie direkt von dem in der Anforderung angegebenen minimalen und maximalen Bedarf ab. Durch die gleich großen Anteile sind kleinere Budgets schneller gefüllt als große. Diese Bevorzugung verstößt jedoch gegen die ursprüngliche Forderung nach Fairness bei der Verteilung, bei der kein Budget unbeabsichtigt einen größeren Anteil erhalten soll.

Die Verteilung auf Grundlage des Anteils am Gesamtbedarf füllt jedes Budget individuell anhand des Bedarfs. Da die minimale Energiemenge bereits verteilt wurde, basiert die zugewiesene Energie auf dem jeweiligen Anteil von  $\max - \min$  am insgesamt verbleibenden Bedarf. Somit werden alle Budgets prozentual gleich gefüllt und die gewünschte Fairness erreicht.

Werden einzelnen Budgets bei der Verteilung durch Gewichte priorisiert, verändert dies die zugewiesene Energiemenge. Eine solche Veränderung der Zuteilung setzt jedoch eine eventuell vorhandene Fairness außer Kraft. Abbildung 5.13 stellt die Auswirkung verschiedener Prioritätszuteilungen auf die durch die Heuristiken verteilte Energiemenge beispielhaft dar.

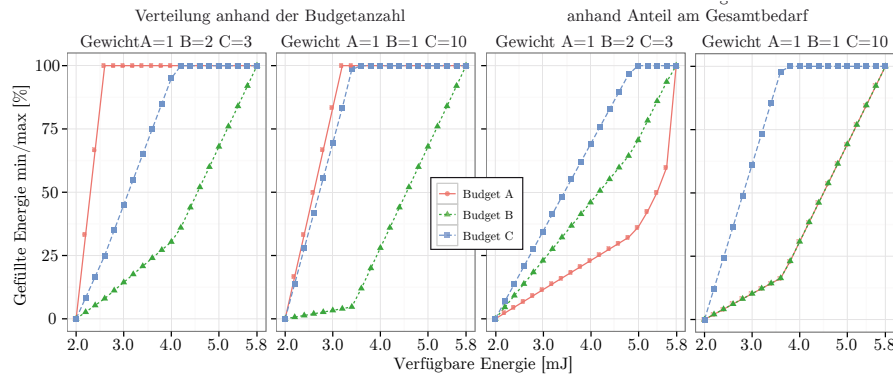


Abbildung 5.13: Beispielhafte Auswirkung von Gewichten auf die Energieverteilung

Die Auswirkungen der Gewichte sind bei der Verteilung anhand gleich großer Anteile nur gering. Die Anteile der Budgets B und C ändern sich. Auf Budget A hat das geringe Gewicht jedoch nur einen geringen Einfluss. Dies ist durch den hohen minimalen Bedarf und vergleichsweise geringen Maximalbedarf der assoziierten Anwendungsteile dieses Budgets begründet.

Bei der Verteilung anhand der Anteile eines Budgets am Bedarf ist der Einfluss der Gewichte deutlich erkennbar. Besitzen alle Budgets unterschiedliche Gewichte, wirkt sich das entsprechend auf die Zuteilung aus. Die Höhe des Gewichtes ist dabei, wie in der Abbildung dargestellt, ebenso von Bedeutung. Je höher dieses ist, desto höher der relative Anteil. Wird nur ein Budget bevorzugt, gilt zwischen Budgets mit gleichem Gewicht weiterhin Fairness.

### 5.6.3 Unzureichende Systemenergie

Abbildung 5.14 zeigt das durch Simulation ermittelte Verhalten verschiedener Strategien, wenn die Systemenergie unzureichend ist, also immer dann, wenn  $E_{avail}(t) < \sum min$  gilt. Die Basis der Simulationen sind die Budgetanforderungen der Evaluation der Energieverteilung. Dabei steht in jedem der simulierten Intervalle nur  $0,5 * \sum min = 1mJ$  Energie zur Verfügung.

Besteht eine Möglichkeit, die Batterien auszutauschen, können alle Budgets zumindest bis zu ihrem jeweiligen  $min$  gefüllt werden. Wie die Abbildung zeigt, reicht die Energie nur aus, um zwei Intervalle lang die minimalen Anforderungen der drei Budgets zu befriedigen. Für zwei weitere Intervalle kann das Budget C mit der höchsten Priorität noch mit Energie versorgt werden.

Strategien basierend auf dem Gewicht verhalten sich unterschiedlich hinsichtlich der Verteilung des Rests, nachdem die Budgets mit dem höchsten Gewicht bis zu ihrem jeweiligen  $min$  gefüllt worden sind. Dieser Rest kann auf die Budgets verteilt werden, deren minimale Anforderungen bereits befriedigt werden konnten. In diesem Beispiel erhält so Budget C die gesamte in diesem Intervall verfügbare Energie.

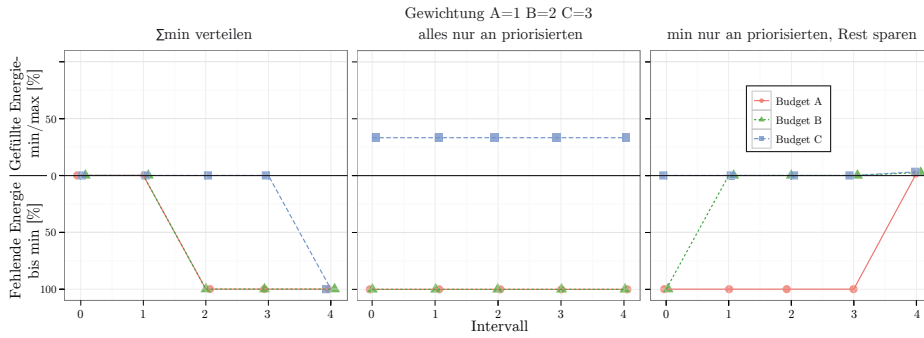


Abbildung 5.14: Verteilung bei unzureichender Energieverfügbarkeit: *min* kann trotzdem verteilt werden. Alternativ kann die Gewichtung genutzt werden, wobei nur priorisierte Budgets gefüllt werden oder die restliche Energie angespart wird.

Der Rest kann auch für spätere Intervalle gespart werden, um später möglicherweise weitere Budgets füllen zu können. Dies führt im simulierten Szenario dazu, dass ab dem zweiten Intervall Budget B ebenfalls Energie erhält. Im letzten Intervall ist dann so viel Energie angespart worden, dass wieder die normale Verteilung verwendet wird.

#### 5.6.4 Strategien für Mehr- und Minderbedarf

Einem kurz- oder langfristigen Mehr- oder Minderbedarf kann auf mehrere Arten begegnet werden. Die Auswirkungen sind, wie bereits in Kapitel 5.4.2 dargestellt, je nach Strategie unterschiedlich. Im Folgenden werden die Möglichkeiten näher untersucht, die das Leihen und Stehlen bieten. Zusätzlich wird das Verhalten bei der Limitierung des Ansparens näher betrachtet.

##### Leihen von Energie

Wird ein Mehrbedarf durch Leihen vom System gedeckt, entsteht für das Budget eine zukünftige Belastung in Form der Kreditrate. Diese beschränkt die durch Leihen verfügbare Energiemenge. In einem ersten Schritt wird deshalb das Anwendungsverhalten bei der Limitierung der durch Leihen beschaffbaren Energie demonstriert, im zweiten der Einfluss der Energiequelle, aus der die geliehene Energie stammt.

Abbildung 5.15 stellt dar, wie sich die Anwendung und das zugrundeliegende Budget verhalten, wenn das Leihen beschränkt wird. In jedem der simulierten 11 Intervalle stehen  $\sum min + 0,5 * (\sum max - \sum min) = 3,9 \text{ mJ}$  zur Verfügung. Der mit Budget A assoziierte Verbraucher benötigt 20% mehr Energie, die zugeteilte Energiemenge reicht also nicht zur Deckung des minimalen Bedarfs aus (0,95mJ zugeteilt vs. minimal 1,08mJ benötigt). Wie die Abbildung zeigt, ist keine der beiden Strategien in der Lage, den konstanten Mehrbedarf über die gesamte Laufzeit zu befriedigen.

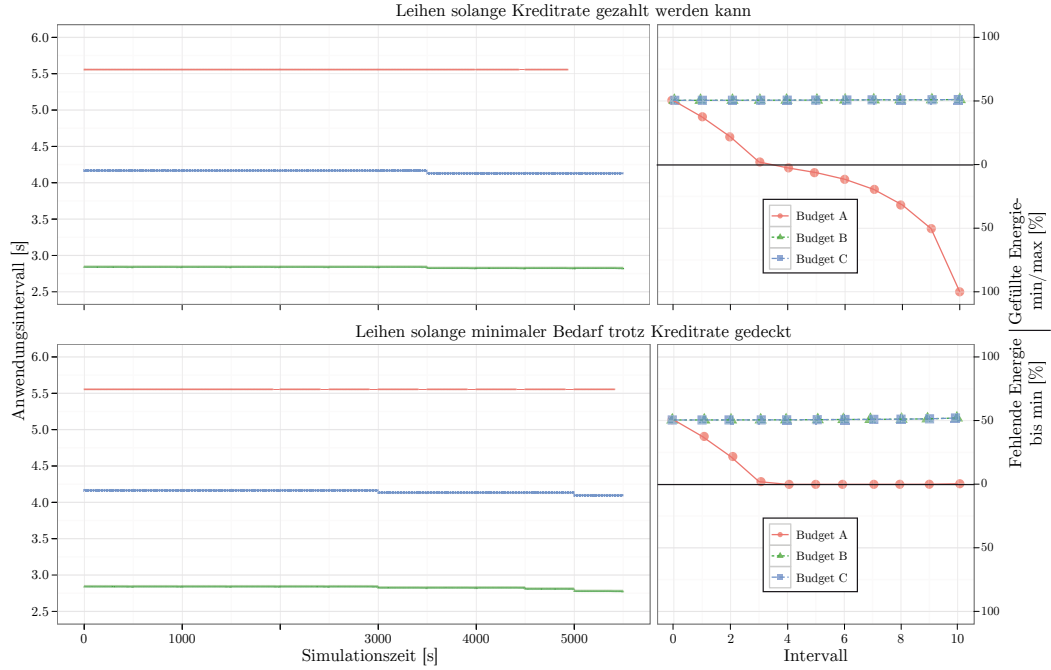


Abbildung 5.15: Einfluss des unlimitierten und limitierten Leihens auf das Anwendungsintervall und die Energiezuteilung, wenn Budget A konstant zu wenig Energie erhält

Wird die leihbare Energiemenge nur durch die Zahlungsfähigkeit der Kreditrate beschränkt, kann der Bedarf des mit dem Budget assoziierten Verbrauchers bis kurz vor Ende der Laufzeit nahezu vollständig befriedigt werden. In jedem Intervall steigt die geliehene Energiemenge und somit die Aufwendungen für die Tilgung. Da im letzten Intervall keine Energie mehr geliehen werden kann, kann der mit A assoziierte Anwendungsteil in diesem Intervall nicht mehr arbeiten (erkennbar an der unterbrochenen Linie des Anwendungsintervalls). Wird die geliehene Energiemenge so beschränkt, dass *min* des Budgets auch nach Abzug der Kreditrate noch befriedigt ist, kann nur ein Teil des Mehrbedarfs gedeckt werden. Nachdem das Limit erreicht wurde, kann keine weitere Energie geliehen werden und die Anwendung kann nicht mehr vollständig arbeiten.

Die Wahl der Strategie hängt vom Ziel der Anwendung und der Ausbringung ab. Ist eine Unterbrechung der Anwendung keinesfalls zulässig, kann das Leihen den Betrieb durch Ausgleichen des Fehlbedarfs für eine gewisse Zeit sicherstellen. Persistiert der Mehrbedarf, kann durch diesen Mechanismus allein der betroffene Anwendungsteil nicht die gesamte Laufzeit arbeiten. Beschränkt man das Leihen und stellt die Deckung des minimalen Bedarfs sicher, ist in diesem Szenario mit konstantem Mehrbedarf kein durchgängiger Betrieb von Anwendungsteil A möglich. Allerdings wird zumindest in den ersten Intervallen der Mehrbedarf gedeckt. Gleichzeitig kann der assoziierte Anwendungsteil mit Unterbrechungen bis zum Erreichen des Lebenszeit-

ziels arbeiten. Diese Strategie stellt somit einen Kompromiss zwischen Isolation und nahezu unbeschränktem Leihen dar und verhindert eine Überschuldung.

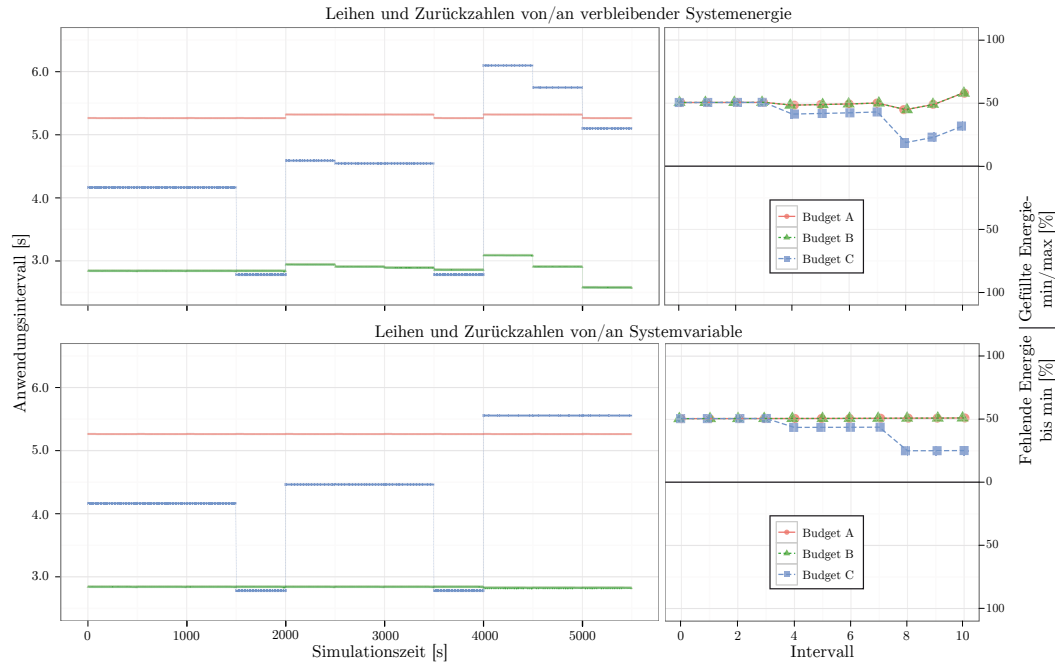


Abbildung 5.16: Einfluss des Leihens von verbleibender Energie oder von einer Systemvariable auf das Anwendungsintervall und die Energiezuteilung, wenn Budget C phasenweise mehr Energie als zugeteilt benötigt

In Abbildung 5.16 wird der Einfluss der Quelle der verliehenen Energie dargestellt. Alle assoziierten Anwendungsteile besitzen den in Tabelle 5.3 angegebenen minimalen und maximalen Bedarf. Die mit Budget C assoziierte Anwendung fordert in diesem Szenario zwei Intervalle lang den maximalen Bedarf ein. Da dieser nicht gedeckt ist, musste für Budget C der Mehrbedarf durch Leihen gedeckt werden.

Wird die fehlende Energie von der verbleibenden Systemenergie geliehen, hat dies einen Einfluss auf alle Budgets. Da weniger Energie verfügbar ist, erhalten alle Budgets einen kleineren Anteil, wodurch die Anwendungsintervalle sinken. Budget C zahlt in den folgenden Intervallen einen Teil der geliehen Energiemenge zurück, wodurch die verbleibende Systemenergie wieder leicht ansteigt. Auf diese Weise wird die geliehene Energie bei der Rückzahlung auf alle Budgets aufgeteilt.

Wird die geliehene Energie in einer Systemvariable protokolliert, hat dies auf die anderen Budgets keinerlei Einfluss. Die Anwendungsintervalle bleiben so bei Budget A und B stabil. Es kommt jedoch zu einer Abweichung der verteilten von der tatsächlich verbrauchten Energie. Dies muss bei der Berechnung der verbleibenden Systemenergie berücksichtigt werden. Durch die Stabilität des Restsystems ist das Leihen von der Systemvariable besser geeignet, um einen Mehrbedarf zu decken.



## Stehlen von Energie

Wird ein Mehrbedarf durch Stehlen von Energie aus anderen Budgets gedeckt, müssen sich die assoziierten Anwendungsteile an die verringerte Energieverfügbarkeit anpassen. Abbildung 5.17 zeigt, welchen Einfluss die Wahl des bestohlenen Budgets hat. Das verwendete Szenario wurde gegenüber dem Leihen nicht verändert. Der mit Budget C assoziierte Anwendungsteil hat einen Mehrbedarf während zweier Intervalle. Das erste dieser Intervalle ist in der Abbildung dargestellt. Die von Budget B und C stehlbare Energiemenge reicht auch kombiniert nicht zur vollständigen Deckung des Bedarfs des Anwendungsteils C aus.

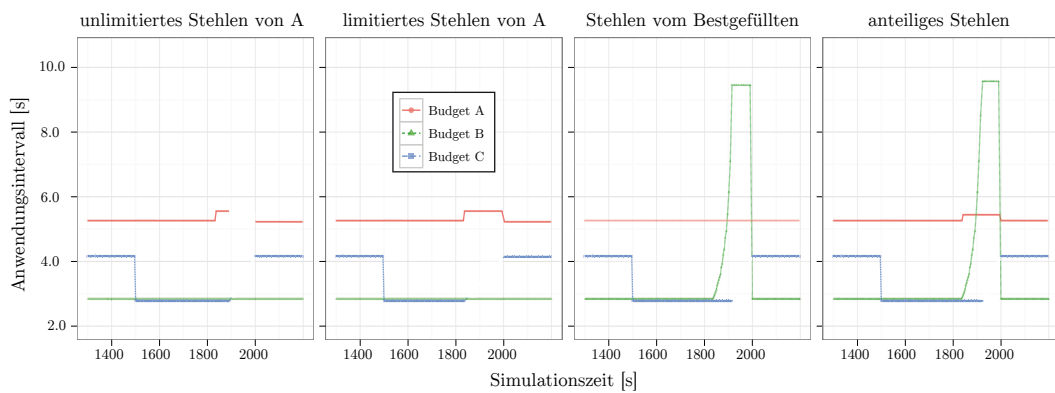


Abbildung 5.17: Einfluss der Strategien zum Stehlen von Energie, um einen Mehrbedarf zu decken: Das Spenderbudget kann fix sein, vom System ausgewählt werden oder mehrere Budgets geben anteilig Energie ab. Wird die Energie einem Budget ohne Limit entzogen, kann der assoziierte Anwendungsteil selbst nicht mehr arbeiten.

Wird das Budget vom Entwickler festgelegt, muss damit gerechnet werden, dass dieses kaum Energie abgeben kann, wie Budget A in diesem Beispiel. Ohne Beschränkung der entnommenen Energiemenge kann das bestohlene Budget selbst nicht mehr die Arbeitsfähigkeit des assoziierten Anwendungsteils aufrechterhalten. Wird die entnehmbare Energiemenge durch den zeitabhängigen Minimalbedarf beschränkt, ist zumindest der minimale Betrieb des bestohlenen Anwendungsteils sichergestellt.

Wählt das System das zu bestehlende Budget aus, kann es das jeweils bestgefüllte Budget wählen. In diesem Beispiel wird jedes Mal Budget B ausgewählt, sodass das Ausführungsintervall von Anwendungsteil B deutlich bis zum Maximum steigt.

Werden mehrere Budgets zur Deckung des Mehrbedarfs herangezogen, geben diese jeweils einen Anteil auf Grundlage ihrer verfügbaren Energie und der benötigten Energiemenge ab. Die Last wird somit auf mehrere Budgets verteilt und die Wahrscheinlichkeit erhöht, Energie zu erhalten.

### Kombination von Leihen und Stehlen

Das Leihen von Energie und das Stehlen können gemeinsam genutzt werden, um Nachteile zu umgehen oder zumindest zu verringern. Eine sinnvolle Kombination ist es, zuerst Energie zu leihen. Dies hat im Gegensatz zum Stehlen keinen Einfluss auf andere Budgets, ist jedoch begrenzt möglich. Kann keine weitere Energie geliehen werden, wird der Mehrbedarf durch Stehlen gedeckt.

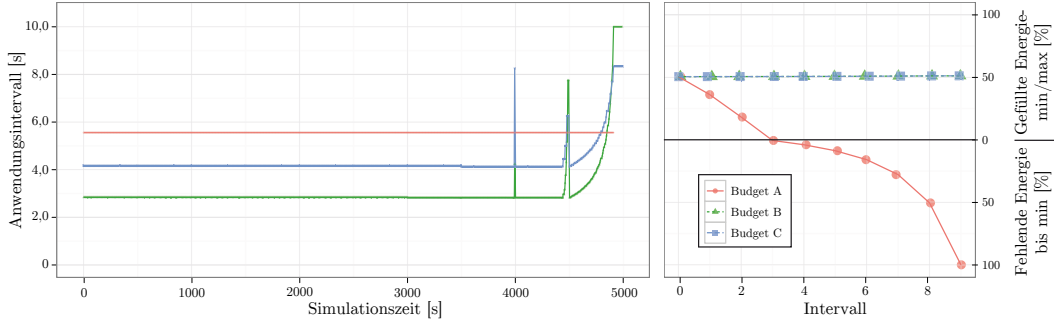


Abbildung 5.18: Auswirkungen auf Anwendungsintervalle und Energieverteilung, wenn der Mehrbedarf erst durch Leihen und anschließend, wenn nötig, durch Stehlen gedeckt wird

Abbildung 5.18 stellt dies für ein Szenario dar. Dieses wurde bereits für die Betrachtung des Leihens verwendet. Anwendungsteil A hat darin einen konstanten Mehrbedarf von 20% durch höhere Kosten. Nachdem Budget A die gesamte zugewiesene Energie für die Tilgung des Kredits aufwenden musste, greift das Stehlen. Es beschafft von den Budgets B und C die notwendige Energie. Im Gegensatz zum reinen Stehlen wird so der Einfluss auf die anderen Budgets minimiert.

#### 5.6.5 Limitierung der angesparten Energie

Wird das Ansparen von Energie zugelassen, kann die notwendige Limitierung statisch oder dynamisch umgesetzt werden. Je nach Höhe des Limits können andere Budgets mehr oder weniger von der überschüssigen Energie profitieren.

Abbildung 5.19 stellt den Einfluss des Limits auf die zugeweilte Energiemenge dar. Im Gegensatz zu den vorangegangenen Simulationen verwendet der Anwendungsteil, der mit Budget C assoziiert ist, nur Energie in Höhe seines minimalen Bedarfs. Um den Effekt des Ansparens zu verdeutlichen, wird der Anteil jedes Budgets statt aus dem Verhältnis von verfügbarer zu maximal benötigter Energie aus dem Verhältnis von minimal zu maximal benötigter Energie bestimmt. Die nach der Deckung des minimalen Bedarfs zugeweilte Energiemenge ist somit unabhängig von der bereits vorhandenen Energie. Zur Verdeutlichung der Strategien wurde das statische Limit für das Ansparen auf 20% des maximalen Bedarfs und  $\kappa$  zur Berechnung des dynamischen Limits  $C$  (siehe Formel 5.15 auf Seite 136) auf 10% festgelegt.

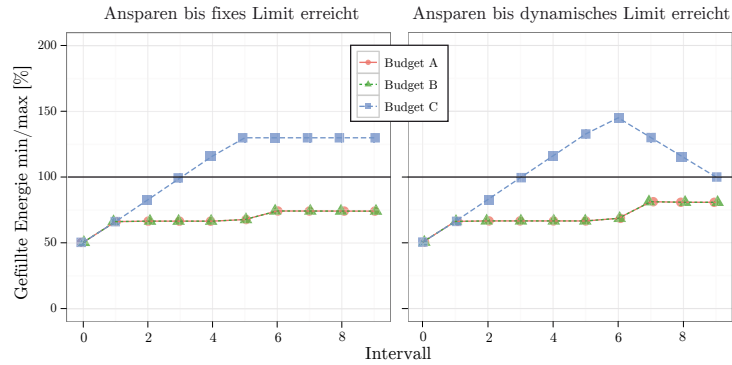


Abbildung 5.19: Einfluss von statischer und dynamischer Limitierung beim Ansparen von überschüssiger Energie: Dargestellt wird der prozentuale Anteil der zugeteilten Energie, nachdem *min* aller Budgets bereits erreicht wurde. 100% entspricht dem jeweiligen *max*.

Bei beiden Strategien steigt im zweiten Intervall für A und B die verfügbare Energiemenge. Dies ist in der Verteilungsstrategie begründet. Budget C erhält, da es ausreichend gefüllt ist, keine Energie zur Deckung des minimalen Bedarfs. Von der so freigewordenen Energie profitieren die Budgets A und B.

Ein statisches Limit für das Ansparen gilt für die gesamte Laufzeit. Ist es erreicht, wird die überschüssige Energie den anderen Budgets zur Verfügung gestellt. Wird die angesparte Energie auch im letzten Intervall nicht verbraucht, bleibt diese ungenutzt und ist verschwendet.

Das dynamische Limit sinkt mit voranschreitender Laufzeit. Zu Anfang der Laufzeit kann viel Energie aufgenommen werden, da diese über die gesamte Restlaufzeit aufgebraucht werden kann. Gegen Ende der Laufzeit kann nur noch wenig angesparte Energie tatsächlich aufgebraucht werden, sodass nicht mehr benötigte Energie den anderen Budgets zur Verfügung gestellt wird.

### 5.6.6 Langfristiger Fehlbedarf

Die Anpassung an einen langfristigen Fehlbedarf ist durch die bisher betrachteten Mechanismen nur eingeschränkt möglich. Zwar kann ein Mehrbedarf durch Leihen oder Stehlen von Energie teilweise befriedigt werden, jedoch hat dies zum Teil drastische Auswirkungen. Ein Minderbedarf kann zwar durch eine Limitierung der überschüssigen Energie kompensiert werden, jedoch bleibt potenziell immer Energie ungenutzt. Eine Anpassung an einen langfristigen Fehlbedarf ist durch die Verteilung von Energie auf Grundlage des tatsächlichen Bedarfs oder die Verwendung von adaptiven Energiebudgets möglich.

## Anpassung der verteilten Energie durch Verteilung des tatsächlichen Bedarfs

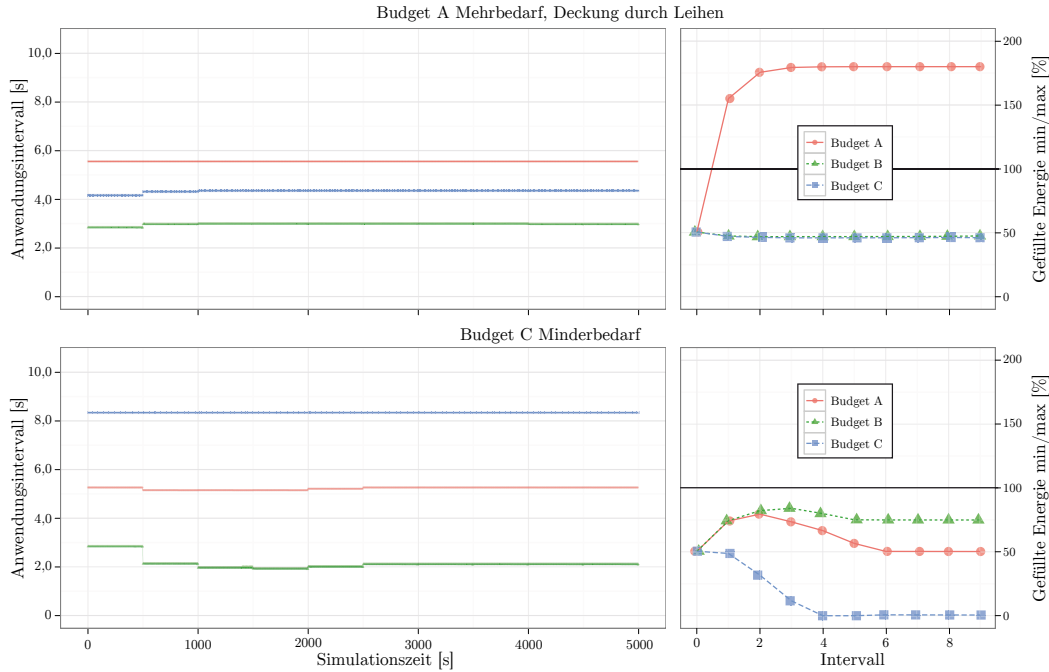


Abbildung 5.20: Auswirkungen der Verteilung des tatsächlichen Bedarfs auf konstanten Mehr- und Minderbedarf

Abbildung 5.20 zeigt, wie sich eine Verteilung des tatsächlichen Bedarfs auf Szenarien mit einem konstanten Mehr- oder Minderbedarf auswirkt. Hierzu wurden die bereits verwendeten Szenarien aus Kapitel 5.6.4 genutzt.

Dem oberen Teil der Abbildung liegt das Szenario zugrunde, bei dem der mit Budget A assoziierte Anwendungsteil einen konstanten Mehrbedarf von 20% hat und dieser durch Leihen gedeckt werden sollte. Im ursprünglichen Szenario war es nicht möglich, den Fehlbedarf über die gesamte Laufzeit zu korrigieren (vergleiche Abbildung 5.15). Zwar wird auch hier Energie geliehen, jedoch erfolgt dies nur während der ersten Intervalle, in denen die Zuteilung zum tatsächlichen Bedarf konvergiert. Dieser liegt über dem ursprünglich angegebenen Maximalbedarf. Eine wie hier erfolgte Zuteilung über den angegebenen Bedarf hinaus sollte nur in geringem Maße erfolgen.

Der untere Teil der Abbildung basiert auf dem Szenario, in dem der mit Budget C assoziierte Anwendungsteil nur den minimalen Bedarf abrufen. Eine Verteilung auf Grundlage des Verbrauchs ermöglicht es, ein Ansparen effektiv zu verhindern und nach der Konvergenz alle nicht benötigte Energie anderen Budgets zuzuteilen (vergleiche Abbildung 5.19). Bis zum Erreichen der Konvergenz erhält Budget C

mehr Energie als benötigt. Im darauf folgenden Intervall wird diese auf die anderen Budgets aufgeteilt.

### Anpassung der verteilten Energie durch Adaption des maximalen Bedarfs

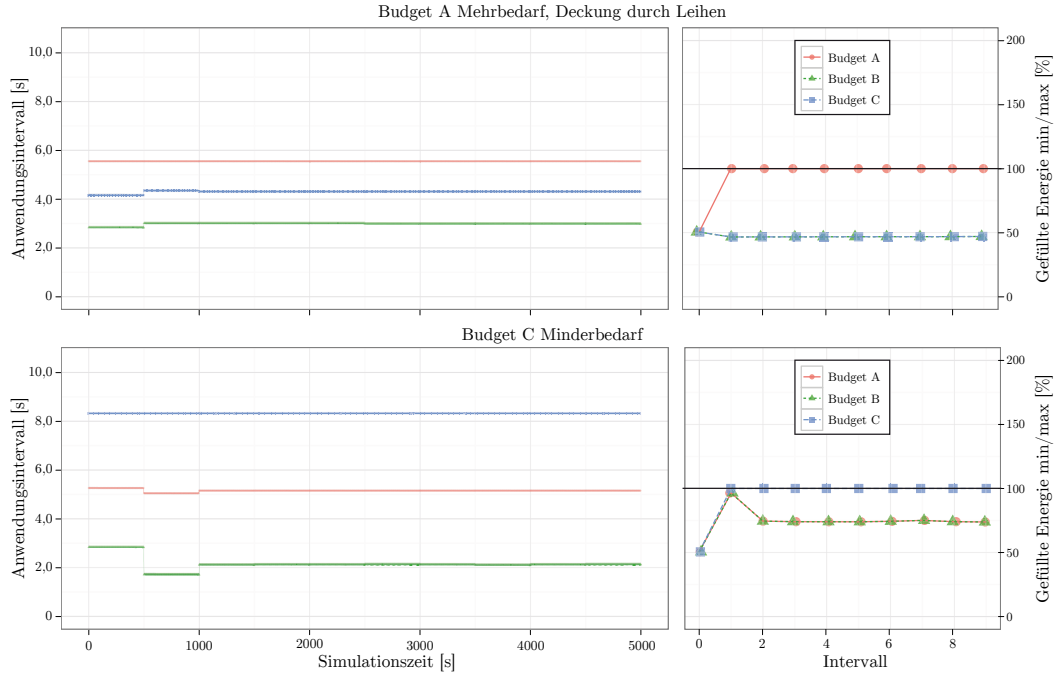


Abbildung 5.21: Auswirkungen der Adaption auf Grundlage des tatsächlichen Bedarfs auf konstanten Mehr- und Minderbedarf

In Abbildung 5.21 ist das Verhalten der Adaption in Situationen mit konstantem Fehlbedarf dargestellt. Die Adaption gestattet es, einen Mehrbedarf durch das System zu decken. Gleichzeitig werden sowohl *min* als auch *max* auf die verbrauchte Energie festgelegt. Dieses Verhalten hilft sowohl beim Mehr- als auch beim Minderbedarf. Im oberen Teil der Abbildung benötigt der mit Budget A assoziierte Anwendungsteil 20% mehr Energie und erhält diese im ersten Intervall vom System. Anschließend werden die Bedarfsangaben des Budgets an den Verbrauch angepasst, wodurch Budget A 100% seiner geforderten Energie zulasten der beiden anderen Budgets erhält. Im unteren Teil der Abbildung verbraucht der mit Budget C assoziierte Anwendungsteil nur den minimal angegebenen Bedarf, wodurch die verteilte Energie von A nach dem ersten Intervall angepasst wird. Da in A noch Energie vorhanden ist, benötigt es in diesem Intervall weniger Energie, wovon Budget A und B profitieren. Danach stabilisiert sich die verbrauchte und verteilte Energie.

Tabelle 5.4: Bewertung der Strategien zur Anpassung an Bedarfsänderungen

Strategie	Variante	Stabilität	Kurzfristig		Langfristig	
			Mehrbedarf	Minderbedarf	Mehrbedarf	Minderbedarf
Isolation	/	+	-	+	-	++
Leihen	verbleibende Energie	-	++	/	--	/
	Systemvariable	++	++	/	--	/
Stehlen	statisches Ziel	-	+	/	+	/
	Systemwahl	--	++	/	+	/
Kombination	/	o	++	/	+	/
Ansparen	unlimitiert	++	/	++	/	-
	statisches Limit	-	/	++	/	+
	dynamisches Limit	-	/	++	/	++
Adaption	/	-	++	++	++	++
Bedarfszuteilung	/	-	o	-	++	++

### 5.6.7 Zusammenfassung der Strategien

Die vorgestellten Strategien haben unterschiedliche Vor- und Nachteile und verschiedene Einsatzszenarien. Tabelle 5.4 fasst die Strategien zusammen und bewertet sie hinsichtlich ihres Einflusses auf andere Budgets und ihrer Fähigkeit, Mehr- und Minderbedarf zu behandeln. Die Bewertung erfolgt dabei nach einem mehrteiligen Maßstab: sehr gut (++), gut (+), neutral (o), schlecht (-) und sehr schlecht (--).

Die reine Isolation ermöglicht keinerlei Deckung eines Mehrbedarfs, schränkt aber gleichzeitig die ansparbare Energiemenge ein und senkt somit potenziell die ungenutzte Energiemenge.

Wird geliehene Energie mithilfe einer Systemvariable protokolliert, hat dies im Gegensatz zum Leihen von der verbleibenden Energie auf die anderen Budgets keinerlei Einfluss. Durch diese Stabilität des Restsystems ist das Leihen von der Systemvariable besser geeignet, um einen Mehrbedarf zu decken. Leihen wird beschränkt durch die Tilgungsfähigkeit und eignet sich daher besser für die Deckung eines kurzfristigen Mehrbedarfs.

Soll Energie eines anderen Budgets gestohlen werden, hat dies einen Einfluss auf die Arbeitsfähigkeit der betroffenen Anwendungsteile und muss soweit beschränkt werden, dass diese weiterhin arbeitsfähig sind. Die Entscheidung, welches Budget bestohlen wird, sollte vom System getroffen werden. So kann die Wahrscheinlichkeit, dass genug Energie verfügbar ist, erhöht werden. Die Aufteilung auf mehrere Budgets erhöht diese Wahrscheinlichkeit weiter. Das Stehlen kann jedoch nur dann den Bedarf

vollständig decken, wenn in den anderen Budgets genügend Energie verfügbar ist. Reicht diese nicht aus, muss der assoziierte Anwendungsteil pausiert werden.

Werden Leihen und Stehlen kombiniert, erfolgt erst dann eine Beeinflussung anderer Budgets, wenn keine weitere Energie geliehen werden kann und gestohlen werden muss. Der Einfluss auf andere Budgets wird somit bei einem kurzfristigen Mehrbedarf minimiert und tritt erst auf, wenn der Mehrbedarf persistiert. Wie beim reinen Stehlen wird die Arbeitsfähigkeit der betroffenen Anwendungsteile von der aus anderen Budgets entnehmbaren Energie bestimmt.

Um zukünftigen Mehrbedarf zu decken, kann Energie angespart werden. Diese wird jedoch anderen Budgets vorenthalten. Erfolgt das Ansparen unbegrenzt, hat dies keine unmittelbaren Auswirkungen auf andere Budgets. Gleichzeitig wird potenziell alle angesparte Energie ungenutzt bleiben. Durch ein Limit wird ein Teil der überschüssigen Energie anderen Budgets zur Verfügung gestellt. Ein dynamisches Limit sinkt mit voranschreitender Laufzeit und macht so alle angesparte Energie nach und nach verfügbar. Im Gegensatz zum statischen Limit bleibt potenziell weniger Energie ungenutzt.

Adaption ermöglicht durch einen Zugriff auf die dem System zur Verfügung stehende Energie eine unmittelbare Anpassung, welche sich jedoch auf andere Budgets auswirkt. Ein Fehlbedarf lässt sich so wirkungsvoll korrigieren. Durch die Anpassung kann es jedoch zu Starvation kommen, wenn ein Budget durch Adaption die gesamte Energie beansprucht. Dies macht eine Limitierung und Kontrolle durch das System oder den Nutzer notwendig. Adaption sollte deshalb nur für wesentliche Anwendungsteile genutzt werden, deren Funktion zwingend, aber deren Verbrauch nicht exakt vorhersagbar ist.

Die Verteilung auf Grundlage des Bedarfs ermöglicht nur eine langsame Anpassung und benötigt zusätzlich weitere Maßnahmen zur Deckung eines initialen Mehrbedarfs. Im Gegensatz zur Adaption werden die angegebenen Bedarfswerte nicht notwendigerweise korrigiert, sondern die Verteilung zugunsten eines Budgets verändert. Starvation ist somit ausgeschlossen.

### 5.6.8 Systemverhalten im Vergleich mit Energy Levels

Um das Verhalten des Managements zu evaluieren, wurde ein experimenteller Vergleich der Energiebudgets mit dem Ansatz der *Energy Levels* [21] durchgeführt. Die auf FeuerWhere-Knoten laufende Anwendung sendet Nachrichten zu einer Senke und hat vier Anwendungsstufen, welche sich jeweils hinsichtlich der Frequenz der Datenerfassung unterscheiden. Der Verbrauch in den Stufen liegt zwischen  $100\mu\text{A}$  und  $270\mu\text{A}$ , was mit konventionellen Alkaline-Batterien Laufzeiten zwischen 32 und 11 Monaten ermöglicht. Periodisch wird die Stufe für das folgende Intervall berechnet. Zur Verbesserung der Vergleichbarkeit basiert die Implementierung aller Ansätze

auf der gleichen Verbrauchserfassung. Zusätzlich wird auf den in [21] beschriebenen Sicherheitsfaktor verzichtet. Somit steht dem System in allen Varianten von Anfang an die gesamte Energie zur Verfügung. *Energy Levels* nutzt einen Simplex Algorithmus<sup>1</sup>, um eine praktikable Zuweisung der Stufen zu berechnen. Hierzu werden die verfügbare Energie, der Verbrauch der vier Stufen und der Nutzen jeder Stufe (1-4) berücksichtigt. Wie in [21] beschrieben, wird nur die höchstmögliche Stufe verwendet, um die größtmögliche QoS zu gewährleisten. Die Variante der *Budgetbasierten Stufen* nutzt das in Kapitel 5.5.2 beschriebene Verfahren, um auf Grundlage des Budgets die mögliche Anwendungsstufe zu berechnen. Zusätzlich wird der Ansatz der Stufen mit einer weiteren Variante verglichen, die direkt das Anwendungsintervall anhand des in Kapitel 5.5.1 beschriebenen Verfahrens bestimmt. Die Laufzeit der Experimente betrug jeweils 12 Intervalle von je 10 Minuten bei einer initialen Energieverfügbarkeit von 25mJ.

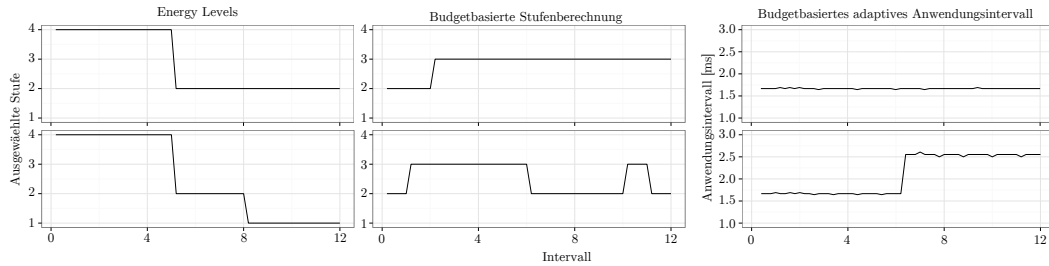


Abbildung 5.22: Experimenteller Vergleich von Managementansätzen: *Energy Levels* berechnet praktikable Stufenzuweisungen, basierend auf der verfügbaren Energie und dem vorgegebenen Nutzen der Stufen. *Budgetbasierte Stufen* bestimmen die aktuelle Servicestufe aufgrund des Budgets. Das *Budgetbasierte Anwendungsintervall* wird anhand der verfügbaren Energie und Kosten eines Intervalls berechnet. Die untere Hälfte zeigt ein Experiment, in dem im 6. Intervall die verbleibende Energie um 20% reduziert wurde.

Die obere Hälfte von Abbildung 5.22 zeigt das Verhalten der drei Varianten. Da *Energy Levels* die höchstmögliche Stufe auswählt, startet es in der höchsten Stufe und wechselt im Laufe der Zeit in die zweitniedrigste. Im Gegensatz dazu wird bei den *Budgetbasierten Stufen* mit der zweitniedrigsten Stufe gestartet, aber kurz darauf in die nächsthöhere Anwendungsstufe gewechselt. Dieser Ansatz ist zwar konservativer, jedoch nicht von der zukünftigen Energieverfügbarkeit abhängig und damit weniger anfällig gegenüber einer Verringerung der verfügbaren Energie, da diese gleichmäßig über die gesamte Laufzeit zugeteilt wird. Das berechnete Anwendungsintervall der dritten Variante ist mit 1,66s stabil und damit etwas höher als Stufe 3 (fixes Intervall von 1,5s).

<sup>1</sup>In dieser Implementierung basierend auf einer Implementierung von J-P Moreau ([www.jpmoreau.fr](http://www.jpmoreau.fr))



Die untere Hälfte von Abbildung 5.22 zeigt den Einfluss einer veränderten Energieverfügbarkeit auf die drei Varianten. In der Mitte des Experimentes wurde die verfügbare Energie um 20% verringert, was beispielsweise durch eine verringerte Batteriekapazität ausgelöst werden kann. *Energy Levels* muss im letzten Drittel des Experimentes auf die tiefste Anwendungsstufe wechseln, da der optimistische Ansatz zu Beginn mehr Energie verbraucht. Die *Budgetbasierten Stufen* erzeugen eine Balance zwischen Energieverbrauch und Laufzeit und müssen das Serviceintervall somit nur um Eins reduzieren, was eine weniger starke Veränderung ist. Das Anwendungsintervall der dritten Variante wird an die veränderte Energieverfügbarkeit auf 2,55s angepasst und ist somit niedriger als Stufe 2 (fix bei 3s).

Beide Varianten, die auf Energiebudgets basieren, bieten im Vergleich nicht nur eine homogenere Anwendungsqualität, sondern möglicherweise kann die Batterie durch die insgesamt niedrigere Last des konservativen Ansatzes von einem verringerten Peukert-Effekt profitieren.

## 5.7 Zusammenfassung

In diesem Kapitel wurde das Konzept der dynamischen Energiebudgets vorgestellt und evaluiert. Durch die Verteilung von „Recht auf Energie“ ermöglichen sie es, den Energiefluss tief eingebetteter Systeme zu steuern und zu limitieren.

Das wesentliche Paradigma der Abgrenzung zwischen Anwendungen und ihren Budgets durch Isolation wurde um die Kooperation ergänzt, um Situationen zu lösen bei denen die zugeteilte Energie unzureichend ist oder ungenutzt bleibt. Hierzu wurden Strategien zur Verteilung der Energie auf unterschiedliche Anwendungsteile bei unterschiedlichen Energieverfügbarkeiten und zum Umgang mit Mehr- und Minderbedarf vorgestellt und auf ihre Eignung in verschiedenen Szenarien hin untersucht.

Im Gegensatz zu existierenden Ansätzen bieten die Strategien mit Mechanismen zum Leihen und Stehlen von Energie Möglichkeiten, auch mit kurzfristigen Veränderungen im Bedarf umzugehen. Auf diese Weise können das Lebenszeitziel und eine möglichst hohe Anwendungsqualität erreicht werden. Eine Anfrage nach mehr Energie kann sowohl explizit von der Anwendung durchgeführt werden, als auch implizit durch das System erfolgen. Anwendungen können somit auch ohne die direkte Einbeziehung von Energie entworfen werden, wodurch die Wiederverwendbarkeit weiter verbessert wird. Statt eines expliziten Energiebewusstseins können sie hierzu die vom System bereitgestellten Funktionen zur Ermittlung von Anwendungsintervallen und Energiestufen nutzen, um sich an eine veränderte Energieverfügbarkeit anzupassen.



# Kapitel 6

## Feldversuche

In diesem Kapitel werden die drei vorgestellten Komponenten des Energiemanagementansatzes zusammengefügt und in zwei Szenarien evaluiert. Das erste Szenario betrachtet das Verhalten des Managements in einem emulierten Industrieumfeld. Parameter eines Gerätes werden überwacht, wobei Ereignisse eines weiteren Sensors zur Steuerung der Erfassung verwendet werden. Im zweiten Szenario wird das Management in einem typischen Sense & Send-Sensornetzwerk für das Umweltmonitoring eingesetzt.

### 6.1 Geräteüberwachung

Die Wartung von Geräten und Anlagen stellt in der Industrie einen erheblichen Kostenfaktor dar. Um diesen zu verringern, wurde im Projekt 4D-Cast<sup>1</sup> mithilfe von Sensoren die tatsächliche Aktivität von Zapfsäulen auf Tankstellen erfasst, um dadurch Rückschlüsse auf deren Verschleiß zu ziehen. Wartungen sollten dann auf Grundlage des tatsächlichen Verschleißes und nicht mehr in festen Intervallen erfolgen, wobei maximale Intervalle definiert wurden.

Das hier umgesetzte Szenario baut auf diesen Gedanken auf und erweitert ihn um das Energiemanagement. Das Management soll eine Laufzeit garantieren, welche über den zu erwartenden Wartungsintervallen liegt. Ein Parameter wird periodisch durch einen Sensor erfasst und direkt an eine Basisstation oder über ein Backbone versendet. Die Rate ist dabei abhängig von der verfügbaren Energie und wird dynamisch nach jedem Sendevorgang bestimmt. Zusätzlich werden externe Ereignisse, wie beispielsweise Bewegung oder Druckveränderung, mittels eines weiteren Sensors erfasst. Tritt ein solches Ereignis auf, wird die Erfassungs- und Senderate kurzfristig deutlich erhöht, um die zeitliche Auflösung der Daten zu verbessern.

---

<sup>1</sup>Bundesministerium für Bildung und Forschung „ZIM-KOOP“ Kooperationsprojekt 4D-Cast „Energiesparende Kommunikationsprotokolle für drahtlose Sensornetze in schwierigen Funkumgebungen mit häufig temporären Funkabschattungen“, Förderkennzeichen: KF2394103JT9

### 6.1.1 Aufbau

Da die exemplarische Umsetzung in dieser Arbeit für die EZ430-Chronos Sensor-knoten erfolgte, wurden die verfügbaren Sensoren genutzt und als Parameter die Temperatur und als Ereignis die Beschleunigung gewählt. Wird eine Beschleunigung festgestellt, wird dem Szenario entsprechend die Erfassungs- und Senderate für zwei Minuten erhöht.

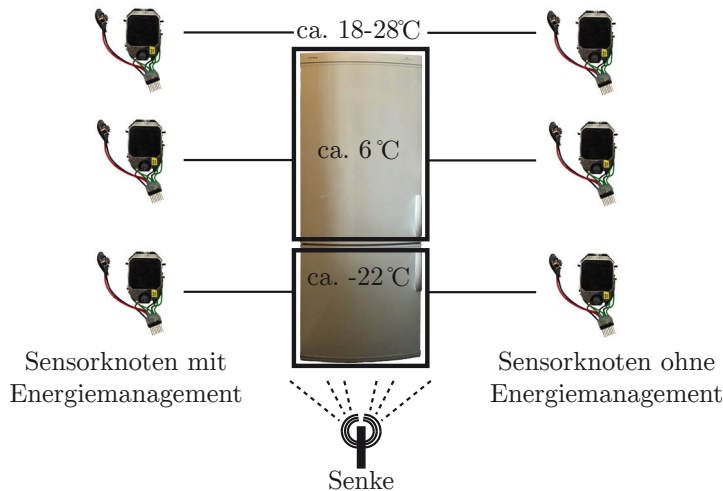


Abbildung 6.1: Aufbau des Szenarios für die Überwachung von Geräten mithilfe drahtloser tief eingebetteter Systeme

Insgesamt wurden sechs Knoten eingesetzt, jeweils drei mit und drei ohne Energiemanagement, um die Temperatur eines Kühlturms zu überwachen (siehe Abbildung 6.1). Dieses Szenario wurde aufgrund der Erreich- und Kontrollierbarkeit gewählt. In Paaren wurden die Knoten auf dem Kühlturm, im Kühlturm und im integrierten Tiefkühlfach positioniert. Durch die Beschleunigungsmessung konnte ein Öffnen oder Schließen des Schrankes von allen Knoten observiert werden.

Das adaptive Energiemanagement passte das Mess- und Sendeintervall an die verfügbare Energie an. Nach der Abarbeitung eines Ereignisses musste das Anwendungsintervall an die verbleibende Energie angepasst werden. Hierzu wurde die verfügbare Energie zum Teil von der Anwendung selbst verwaltet. Ein Teil der Energie wurde von der Anwendung in ein nicht vom Management verwaltetes Ereignisbudget transferiert. Dieses wurde für die Deckung der Energiekosten während der erhöhten Rate bei Ereignissen genutzt. Reicht das Budget nicht aus, wurde notwendige Energie aus dem Anwendungsbudget gestohlen. Da dies beschränkt ist, wie in Kapitel 5.4.2 beschrieben, wurde weiterer Bedarf vom System gedeckt. Ein drittes Budget wurde für die Isolation des Verbrauchs des Beschleunigungssensors und dessen Auswertung verwendet. Da dieser Verbrauch leicht schwankt, wurde ein adaptives Budget verwendet, dessen Größe vom System dynamisch bestimmt wird.

Tabelle 6.1: Parameter des Kühlschrankszenarios

Anwendung	Laufzeit	275h
	Intervall (fix)	1,25 s
	Intervall (min, max)	0,5 s, 60 s
	Intervall (Ereignis)	0,5 s
Batterieüberwachung	T	165 je 100 Minuten
	$B_+, B_-$	0,5 und 0,1
	$V_{cutoff}$	2,25V
	Phase C	ab 2,35V
Energiemanagement	T	1650 je 10 Minuten
	Angenommene Ladung	1050 mAh
	Adaption pro Hinweispunkt	12,5%

Die Parameter des Szenarios sind in Tabelle 6.1 zusammengefasst. Als Energieversorgung wurden Noname Alkaline AAA Batterien genutzt, da diese bei den niedrigen Temperaturen, die im Tiefkühler vorherrschen, grundsätzlich funktionsfähig bleiben (siehe Experimente in Kapitel 3.4). Das fixe Intervall wurde so dimensioniert, dass das Lebenszeitziel bei Umgebungstemperatur mit einem ausreichenden Sicherheitsfaktor erreicht werden konnte. Da die verwendeten Knoten nur über einen geringen Programmspeicher verfügten, wurden für die Budgets 32bit Werte genutzt. Dies beschränkte die Aufnahmefähigkeit der Budgets und machte eine entsprechend hohe Zahl an Intervallen für das Management nötig. Da dies jedoch für die Batterieüberwachung kontraproduktiv ist, wurden beide asynchron voneinander betrieben. Die sonstigen Parameter entsprachen den bereits in anderen Experimenten verwendeten (siehe Kapitel 3.4).

### 6.1.2 Auswertung

Das Ziel, den Knoten eine vorher definierte Laufzeit zu ermöglichen, konnte mithilfe des Energiemanagements in Zusammenarbeit mit der Batterieüberwachung erreicht werden. Zwei der drei Knoten ohne Management konnten das Lebenszeitziel ebenso erreichen, jedoch fiel der Knoten im Tiefkühler bereits im zweiten Drittel des Versuchs aus. Dies zeigt wie schon die Experimente in Kapitel 3.4 den Einfluss der Temperatur auf Batterien. Je tiefer die Temperatur, desto weniger Energie steht zur Verfügung. Im Gegensatz dazu konnte der Einfluss durch das Management mithilfe von Anpassungen des Arbeitsintervalls ausgeglichen werden.

Die Zeitpunkte der Anpassungen der drei Knoten mit Energiemanagement sind in Abbildung 6.2 dargestellt. Auf jedes Hinweissignal des abstrakten Batteriemodells hin wird die verbleibende Energiemenge durch das Energiemanagement angepasst. Insbesondere das Management des Knotens innerhalb des Tiefkühlfaches musste

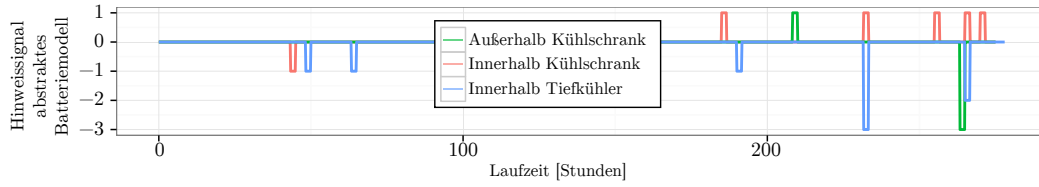


Abbildung 6.2: Hinweissignale des abstrakten Batteriemodells

bereits zu Beginn die verbleibende Energiemenge verringern. Gegen Ende des Experimentes verstärkte sich das negative Feedback noch, wodurch der Energieverbrauch weiter verringert werden musste. Im Gegensatz dazu konnten die anderen Knoten zumindest teilweise ihren Verbrauch erhöhen.

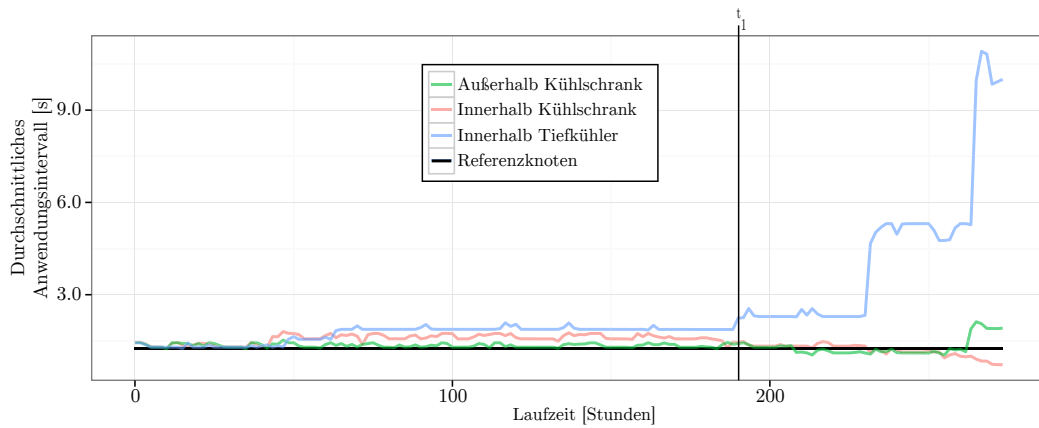

 Abbildung 6.3: Durchschnittliches Anwendungsintervall der Knoten mit Energiemanagement: Zum Zeitpunkt  $t_1$  versagt der Referenzknoten im Tiefkühler.

Abbildung 6.3 zeigt die Auswirkungen der Anpassungen durch die Batterieüberwachung auf das durchschnittliche Anwendungsintervall. Der Knoten innerhalb des Tiefkühlschranks konnte nur durch eine deutliche Einschränkung des Anwendungsintervalls das Lebenszeitziel erreichen. Verglichen mit den Referenzknoten außerhalb des Tiefkühlschranks, wurden etwa 40% weniger Nachrichten versendet. Der Referenzknoten innerhalb des Tiefkühlschranks konnte das Lebenszeitziel nicht erreichen und versagte am Zeitpunkt  $t_1$ . Die Knoten außerhalb und innerhalb des Kühlschranks versendeten etwa 5% und 9% weniger Nachrichten als die Referenzknoten. Ursachen hierfür sind sowohl der Mehraufwand und das konservative Verhalten des Managements als auch die verringerte Energiemenge durch das Ansparen für die Ereignisabarbeitung.

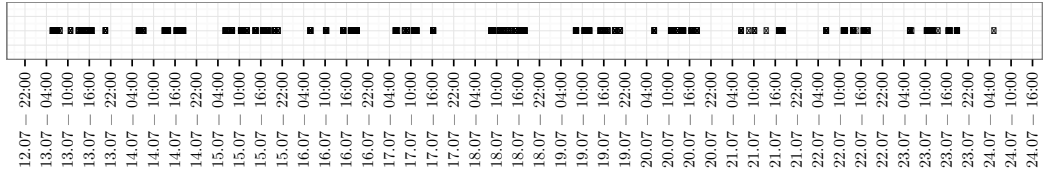


Abbildung 6.4: Zeitliche Verteilung der erfassten Ereignisse

Die Verteilung der aufgetretenen Ereignisse ist in Abbildung 6.4 dargestellt und entspricht den Erwartungen aufgrund der Art der Ereignisse. Alle Knoten konnten das Öffnen beziehungsweise Schließen des Kühlschranks zuverlässig detektieren. Es trat eine Häufung von Ereignissen tagsüber auf, insbesondere morgens und abends. Nachts hingegen wurden keine Ereignisse festgestellt.

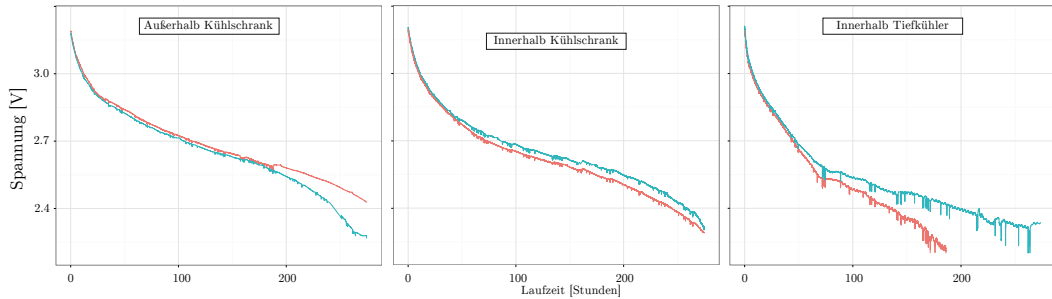


Abbildung 6.5: Gemessener Spannungsverlauf der Knoten außerhalb und innerhalb des Kühlschranks sowie im Tiefkühlfach, sowohl ohne das vorgestellte Management (rot) als auch mit (blau)

Das Auftreten der Ereignisse wirkte sich sowohl auf die Spannung als auch auf das Anwendungsintervall der folgenden Managementintervalle aus. Durch die größere Last während der Abarbeitung der Ereignisse konnte die Spannung kurzfristig einbrechen. Wie in Abbildung 6.5 dargestellt, betraf dies insbesondere die Knoten im Tiefkühlfach. Dies kann potenziell so weit gehen, dass die minimale Arbeitsspannung des Knotens erreicht wird und dieser sich vorzeitig abschaltet. Um dem vorzubeugen, könnte für weitergehende Experimente oder tatsächliche Anwendungen jeweils die Spannung während der Ereignisabarbeitung als Grundlage für die Batterieüberwachung genutzt werden. Dies würde jedoch Energie in der Batterie ungenutzt lassen.

Das Auftreten eines Ereignisses hatte eine Auswirkung auf das normale Anwendungsintervall. Wie in Abbildung 6.6 dargestellt, war dies abhängig vom zugeordneten Budget. War das Ereignisbudget groß genug (links), stand nach dem Auftreten und Abarbeiten des Ereignisses ( $t_1$  bis  $t_2$ ) potenziell mehr Energie im Anwendungsbudget zur Verfügung, da dieses während der Ereignisabarbeitung nicht genutzt wurde und so für die verbleibende Zeit des Managementintervalls ein kleineres Anwendungsintervall möglich war. Mit dem neuen Managementintervall ( $t_3$ ) stieg auch das Anwendungsintervall, da von der Anwendung ein Teil der zur Verfügung ste-

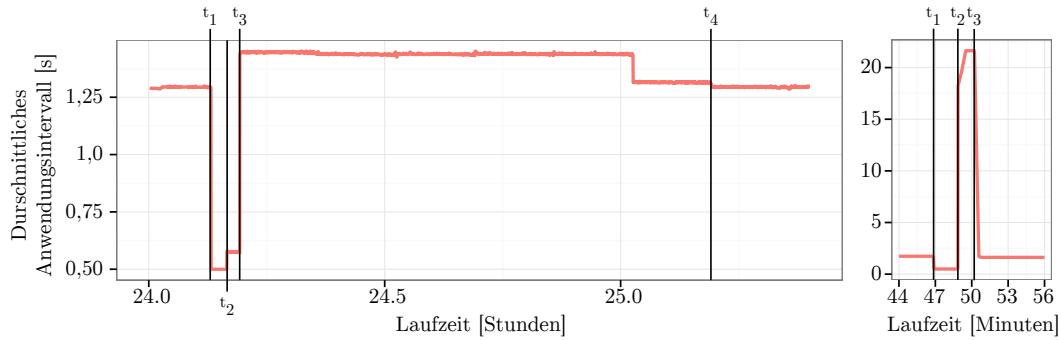


Abbildung 6.6: Mögliche Auswirkungen der Ereignisabarbeitung auf das Anwendungsintervall: Ereignisbudget gefüllt (links) und leer (rechts)

henden Energie in das Ereignisbudget übertragen wurde. Erst wenn dieses wieder vollständig gefüllt war ( $t_4$ ), konnte wieder das ursprüngliche Anwendungsintervall erreicht werden.

Enthielt das Ereignisbudget nicht genug Energie (rechts), wurde der darüber hinausgehende Bedarf vom Anwendungsbudget gedeckt. Dies erhöhte im Anschluss an die Abarbeitung des Ereignisses ( $t_1$  bis  $t_2$ ) das Anwendungsintervall deutlich, welches erst nach der nächsten Zuteilung ( $t_3$ ) wieder auf den ursprünglichen Wert fiel.

Diese Auswirkungen ließen sich durch Anpassen der Anwendung umgehen. So ist es möglich, das Anwendungsintervall nicht mehr nach jedem Sendevorgang neu zu berechnen, sondern nur einmal nach der jeweiligen Zuteilung der Energie. Des Weiteren kann fehlende Energie im Ereignisbudget durch direkte Entnahme aus dem System gedeckt werden, was jedoch gerade am Ende der Laufzeit problematisch sein kann.

Im vorgestellten Szenario ermöglichte das Energiemanagement das Erreichen des Lebenszeitziels trotz des starken Temperatureinflusses. Die Anwendung zeigt, dass mithilfe der Budgets ein Energiebewusstsein erzeugt werden kann und somit die Auswirkungen der Ereignisse auf die Anwendung reduziert werden.

## 6.2 Hydrologisches Umweltmonitoring

Das in der Niederlausitz gelegene Hühnerwasser wird zur Untersuchung eines initialen Zustands eines Ökosystems im ehemaligen Braunkohletagebau genutzt [205]. Dabei werden im 6 Hektar umfassenden künstlichen Wassereinzugsgebiet unter anderem hydrologische Untersuchungen durchgeführt. Traditionell mussten dazu Daten durch Ablaufen von Messpunkten manuell, teils in Papierform, gesammelt werden. Dies stellt auf einem großflächigen Gelände mit vielen Messpunkten wie dem Hühnerwasser einen hohen Aufwand dar und schränkt die erreichbare zeitliche Auflösung ein. Sensornetzwerke können in solchen Szenarien nicht nur den Aufwand verringern



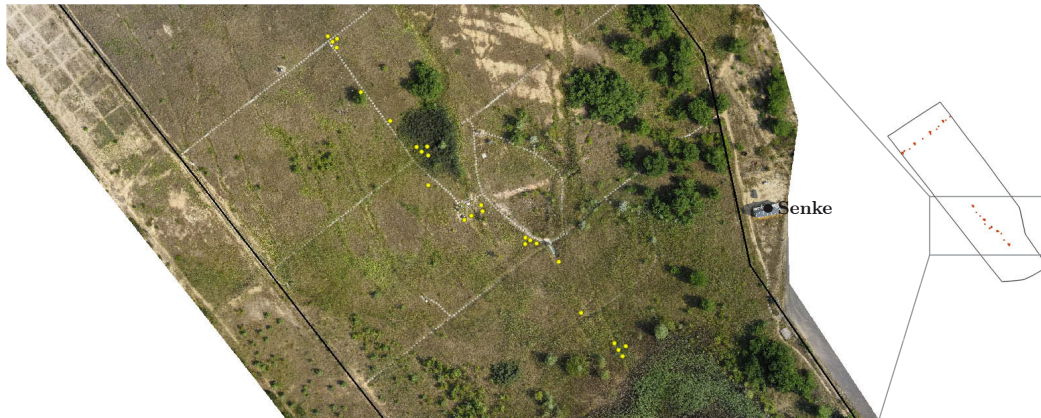


Abbildung 6.7: Überblick über den unteren Teilabschnitt des künstlich angelegten Wassereinzugsgebietes Hühnerwasser mit Positionen der Sensorknoten und der Senke (nach [207] verfeinert)

und die Auflösung erhöhen, sondern auch den menschlichen Einfluss auf das Ökosystem minimieren. Aus diesem Grund wurden im Sense4U Projekt<sup>2</sup> an 50 Messpunkten Sensorknoten mit dem Ziel installiert, hydrologische Daten zu erfassen [206].

Abbildung 6.7 zeigt eine schematische Darstellung des gesamten Geländes und eine Übersicht über den auch für das Experiment in dieser Arbeit verwendeten unteren Teilabschnitt des Hühnerwassers.

Basis der Knoten bildeten FeuerWhere Sensorknoten, die mit Sensorboards über verschiedenste Sensoren verfügen (unter anderem SHT11 [208] für Temperatur und Luftfeuchtigkeit) und in wasserfesten Gehäusen verpackt wurden. Zusätzlich wurden für die Erfassung der Regenmenge Kippwaagen installiert. Diese erzeugen mithilfe eines Reedkontaktes Unterbrechungen auf dem Mikrocontroller, wenn Niederschlag in Höhe von 0,2l /qm gesammelt wurde.

Für hydrologische Untersuchungen ist insbesondere der Einfluss der Vegetation auf die Niederschlagsmenge in Bodennähe von Bedeutung. Die Sensorknoten wurden deshalb in Gebieten mit unterschiedlicher Vegetationsdichte installiert. Abbildung 6.8 zeigt die drei Vegetationsdichten im unteren Teilabschnitt des Geländes.

Die Knoten wurden den beteiligten Hydrologen nach Ende des Projektes weiterhin zur Verfügung gestellt, jedoch wurden sie als reine Datenlogger verwendet, um die laufenden Kosten und den Wartungsaufwand zu verringern (keine Internetverbindung und Stromversorgung nur über Solar und Windkraft möglich). So konnten die Sensorknoten als Experimentierplattform für die Evaluation des vorgestellten Ansatzes mit verwendet werden.

---

<sup>2</sup>Innovative regionale Wachstumskerne mit WK Potential des Bundesministeriums für Bildung und Forschung „Sense4U: Drahtlose Sensornetze zur Verbesserung des Umweltmonitorings“, Förderkennzeichen: 03WKP26B

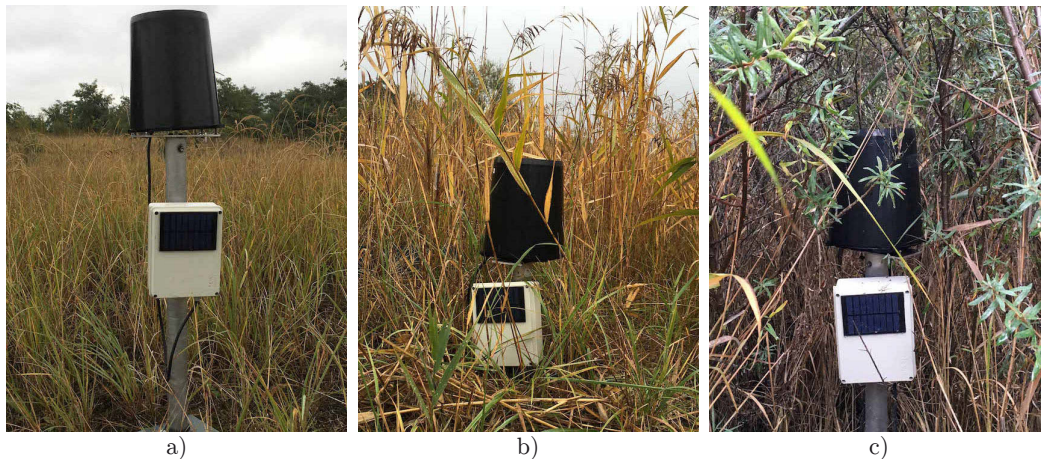


Abbildung 6.8: Zonen mit unterschiedlicher Vegetationsdichte, in denen Knoten im unteren Teilabschnitt ausgebracht wurden: a) offene Fläche, b) Schilf und c) Sanddorn

### 6.2.1 Aufbau

Vom 4. bis zum 20. August 2015 wurde die Eignung des Energiemanagementansatzes im Feld überprüft. Dabei wurden der Niederschlag sowie periodisch Temperatur und Luftfeuchtigkeit erfasst und jeder Wert auf einer SD-Karte gespeichert. Dies ermöglichte die Bereitstellung der Daten in voller zeitlicher Auflösung, was insbesondere für die Regenmenge wichtig war. Zusätzlich wurden Parameter des Energiemanagements und Routings auf der Karte gespeichert.

Da im Grunde jeder gemessene Datenpunkt von Bedeutung und Interesse ist, aber nicht übertragen werden kann, müssen Daten auf einer SD-Karte gespeichert werden. Die Kosten hierfür sind zwar zunächst höher, nach dem möglichen Verzicht auf Synchronisation durch DutyCycle und Datensicherheit durch Retransmissionen ist der Aufwand jedoch deutlich geringer.

Per Funk wurde von jedem Knoten alle 20 Minuten eine Statusnachricht mit aggregierten Sensorwerten versendet. Da der Abstand zwischen den beiden Abschnitten des Hühnerwassers (vergleiche Abbildung 6.7) die Verwendung von Brückenknoten notwendig macht, wurden von den 50 vorhandenen Knoten im Rahmen dieser Arbeit nur die 25 Knoten des unteren Teilabschnittes verwendet. In diesem Teilabschnitt kann die Senke von der Mehrzahl der Knoten meistens direkt erreicht werden, wodurch nur ein einfaches Wegwahlverfahren eingesetzt wurde. Dieses basiert auf der Entfernung zur Senke, welche in den ACK-Nachrichten übermittelt wurde. Jeder Knoten mit einer kleineren Entfernung leitete die Nachricht weiter, es sei denn, er observierte eine Weiterleitung durch einen anderen Knoten. Zur Reduktion des Energieverbrauchs des Funkmoduls wurde das Arbeitsintervall auf 2 Sekunden alle

Tabelle 6.2: Parameter des Umweltmonitoringszenarios

Anwendung	Laufzeit	360 Stunden
	Intervall (Sensor fix)	1,75 s
	Intervall (Sensor min, max)	0,2 s, 60 s
	Intervall (Parametersicherung)	5 Minuten
	Intervall (Kommunikation)	20 Minuten
Batterieüberwachung	T	180 je 2 Stunden
	$B_+, B_-$	0,5 und 0,1
	$V_{cutoff}$	2,0V
	Phase C	ab 2,3V
Energiemanagement	T	180 je 2 Stunden
	Angenommene Ladung	180 mAh
	Adaption pro Hinweispunkt	12,5%

10 Minuten festgelegt. Die Synchronisation erfolgte mithilfe von Zeitstempeln, die ebenfalls in den ACK-Nachrichten übermittelt wurden.

Die Laufzeit des Experimentes wurde durch Verwendung von teilgeladenen Ene-loop AA Akkumulatoren [26] auf 15 Tage verringert. Die Laufzeiten bei der Verwendung von vollgeladenen Standard-Batterien lägen um ein Vielfaches höher, was für ein solches Experiment nicht praktikabel ist. Die verfügbare Ladung von 200mAh wurde um 10% geringer geschätzt, um einen Sicherheitsfaktor zu integrieren.

Tabelle 6.2 fasst die Parameter des Szenarios zusammen. Die Laufzeit des Experimentes wurde in 180 Intervalle mit je 2 Stunden aufgeteilt. Die Intervalle der Batterieüberwachung und des Energiemanagements waren synchronisiert. Erzeugte Hinweise führten direkt zu einer Anpassung der verbleibenden Energiemenge. Anschließend wurde die Energie für das aktuelle Intervall auf die Budgets verteilt.

Es wurden fünf Budgets verwendet. *Kommunikation*, *SD Parameter*, *Niederschlag* und *System* waren adaptive Budgets, da der Energieverbrauch der assoziierten Anwendungsteile schwankte, sie aber essentiell für die Ausbringung waren. *System* bediente den Energieverbrauch der Schlafzustände aller Knotenkomponenten. Zwar ließen sich diese Budgets in einem einzelnen adaptiven Budget zusammenfassen, jedoch ermöglicht die Aufspaltung eine bessere Identifikation der Verbräuche im System und erhöhte die Wiederverwendbarkeit einzelner Komponenten zusammen mit ihren Parametern.

Das *Sensoren*-Budget erhielt alle verbleibende Energie, die in einem Intervall zur Verfügung stand. Der assoziierte Anwendungsteil steuerte den SHT11 für die Messung von Temperatur und Luftfeuchte und nutzte die vom System bereitgestellte Funktionalität zur Anpassung des Messintervalls an die zur Verfügung stehende Energie (siehe Kapitel 5.5.1).

Fünf Knoten wurden als Vergleichsknoten genutzt. Sie enthielten kein dynamisches Energiemanagement und arbeiteten mit einem statischen Messintervall von 1,75 Sekunden, was diesen Knoten das Erreichen des Lebenszeitziels mit ausreichendem Sicherheitsabstand ermöglichte.

### 6.2.2 Auswertung der Ausbringung

Regen trat während der Ausbringung nur an den letzten Tagen auf und war bis auf den 17. August nur nieselig. Abbildung 6.9 zeigt die durchschnittliche erfasste Niederschlagsmenge in den drei Vegetationszonen. Die Werte der offenen Fläche und im Schilf unterschieden sich nur geringfügig, jedoch erfassten die Knoten im Sanddorn erkennbar weniger Niederschlag. Die Knoten im Schilf erfassten 1,8% mehr Regen als die auf offener Fläche, während die Knoten im Sanddorn 16% weniger erfassten.

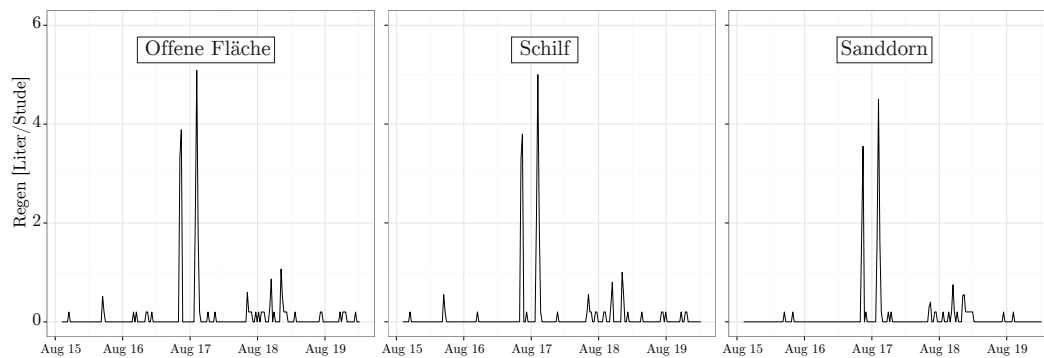


Abbildung 6.9: Durchschnittliche Regenmenge in den drei Vegetationszonen.

Die Temperaturkurve ist in Abbildung 6.10 auf der linken Seite dargestellt. Unterschiede, hervorgerufen durch die Vegetationsdichte und den daraus folgenden Schatten, sind deutlich erkennbar. Knoten auf offener Fläche erfassten eine durchschnittliche Temperatur von  $26,13^{\circ}\text{C}$ . Mit  $24,85^{\circ}\text{C}$  war die durchschnittliche Temperatur im Schilf niedriger. Die durch den Sanddorn verursachten Schatten senkten die gemessene Temperatur auf  $23,66^{\circ}\text{C}$ . Zusätzlich wurde die maximale Temperatur durch die Vegetation beeinflusst und war im Sanddorn deutlich geringer.

Der rechte Teil von Abbildung 6.10 zeigt die durchschnittliche Luftfeuchtigkeit während der Ausbringung. Wieder beeinflusst die Vegetation den Umweltparameter. Die durchschnittliche Luftfeuchte ist bei den Knoten auf offener Fläche am niedrigsten (58,3%). Im Schilf herrschte mit 71,1% eine deutlich höhere Luftfeuchtigkeit. Im Sanddorn war die durchschnittliche Luftfeuchte mit 77,2% am höchsten.

Die Daten zeigen den Einfluss der Vegetationszone auf die erfassten Umweltparameter. Die Regenmenge wird in dichter Vegetation reduziert und gleichzeitig die Luftfeuchtigkeit erhöht.

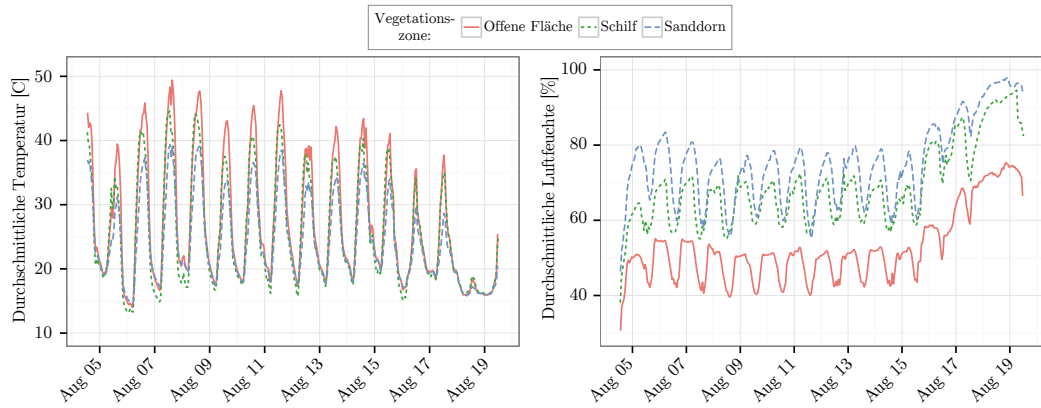


Abbildung 6.10: Durchschnittlich gemessene Temperatur und Luftfeuchte in den drei Vegetationszonen

### 6.2.3 Auswertung des Managements

Nicht alle Knoten haben das Lebenszeitziel erreicht. Vier der Knoten sind abgestürzt (drei waren später ebenfalls problematisch), sodass nur 21 Knoten das Ende des Experimentes erreichten. Die bereits vorher mehrfach verwendeten SD-Karten haben Fehler erzeugt, sodass nur Daten von 18 Knoten ausgewertet werden konnten. In den übertragenen Nachrichten wurden nur eine Zusammenfassung der letzten erfassten Parameter und das aktuelle Managementintervall übertragen. Es standen Daten von 15 Knoten mit Management und drei Vergleichsknoten zur Verfügung.

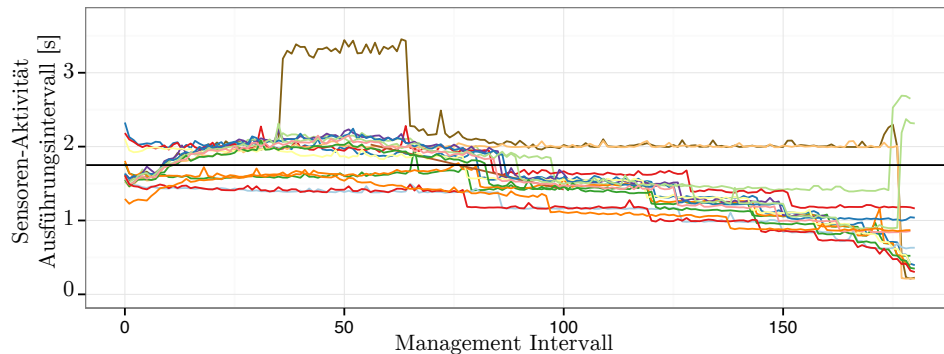


Abbildung 6.11: Berechnetes Messintervall der Sensoren-Aktivität: Die durchgehende schwarze Line markiert das statische Intervall der Vergleichsknoten

Abbildung 6.11 stellt das dynamisch berechnete Messintervall der Sensoren-Aktivität dar. Während des Experimentes konnten die meisten Knoten ihr Intervall verkleinern und so die Datenqualität verbessern. Die Vergleichsknoten erfassten Daten alle 1,75 Sekunden. Das durchschnittliche Messintervall über alle Knoten mit Management betrug 1,61 Sekunden. Der beste Knoten hatte ein durchschnittliches Intervall von 1,16s und der schlechteste von 2,18s. Dieser litt unter Fluktuationen



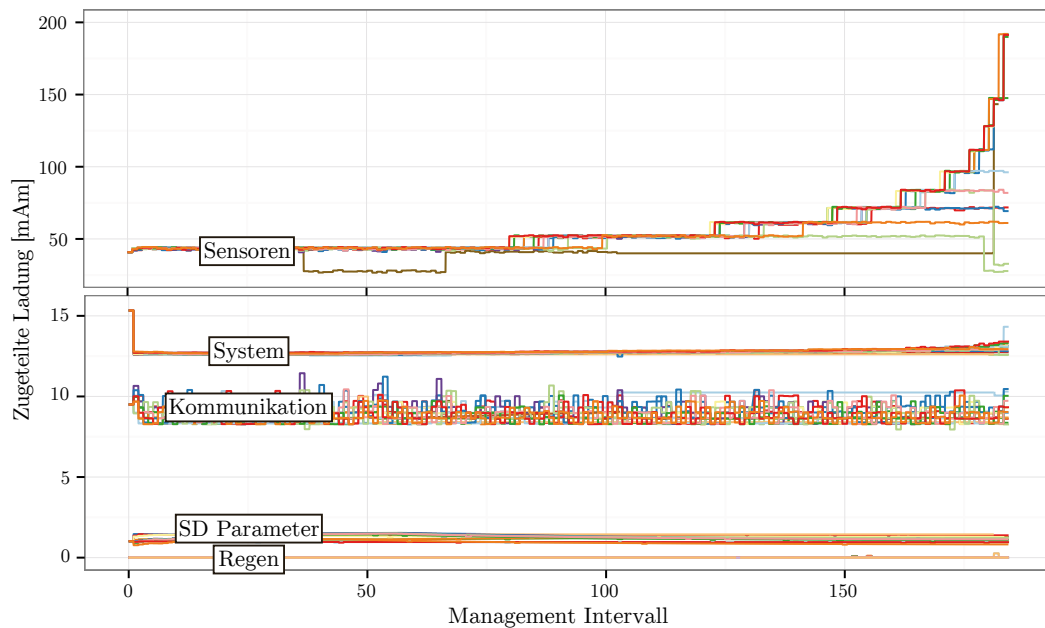


Abbildung 6.12: Größe der Budgets über die Laufzeit

der Batteriespannung (vergleiche hierzu Abbildung 6.14) und verringerte dadurch seine Messfrequenz im ersten Drittel der Laufzeit.

Die Größe der fünf Budgets ist in Abbildung 6.12 dargestellt. Die Kosten für die Kommunikation schwankten, was durch die adaptive Anpassung des Budgets ausgeglichen werden sollte. Dies wirkte sich auf das Sensor-Budget aus, das im selben Maße schwankte. Eine Abschwächung der Adaption, beispielsweise durch Mittelwertbildung, kann diese Schwankungen verringern. Die Steigerung der Größe des Sensoren-Budgets durch die Erhöhung der Energiemenge aufgrund der Hinweise der Batterieüberwachung ist deutlich zu erkennen.

Der durchschnittliche Verbrauch zum Ende der Laufzeit aller Knoten ist in Abbildung 6.13 dargestellt. Die auf einzelne Systembestandteile aufgeschlüsselten Werte wurden zur Laufzeit durch die Verbrauchserfassung ermittelt. Der größte Verbraucher war die SD-Karte. Der Hersteller und das Alter jeder Karte hatten einen Einfluss auf die Schreibkosten und somit direkt auf das Messintervall jedes Knotens. Die Schreibkosten wurden wie in Kapitel 5.3.2 beschrieben durch Erfassung der Energiekosten und der geschriebenen Bytes berechnet. Die benötigte Ladung pro geschriebenem Byte lag zwischen 9,8 und 16mAh.

Abbildung 6.14 zeigt, dass die Batteriespannung der meisten Knoten von der Abschaltspannung 2,0V weit entfernt war. Dies deutet darauf hin, dass die Ladung der Batterien höher war als angenommen. Die meisten Knoten verbrauchten mehr als die ursprünglich geladenen 200mAh. Die Ursache konnte im Ladeprozess der Batterien gefunden werden. Die Batterien wurden vollständig entladen und ruhten im Ladegerät, bevor sie geladen wurden. Das Ladegerät legte während der Ruhephase

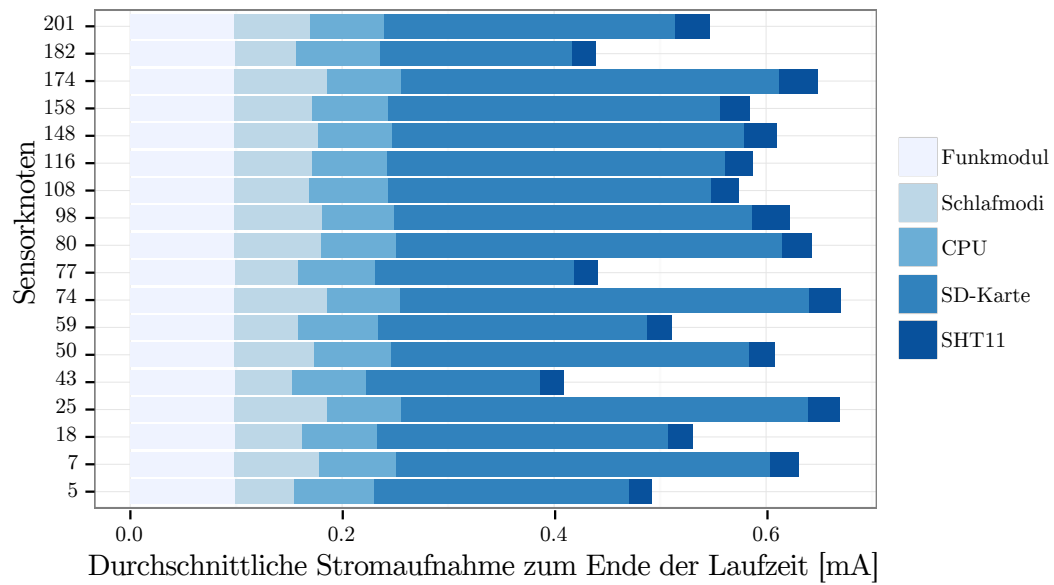


Abbildung 6.13: Von der Verbrauchserfassung ermittelte durchschnittliche Stromaufnahme der Sensorknoten

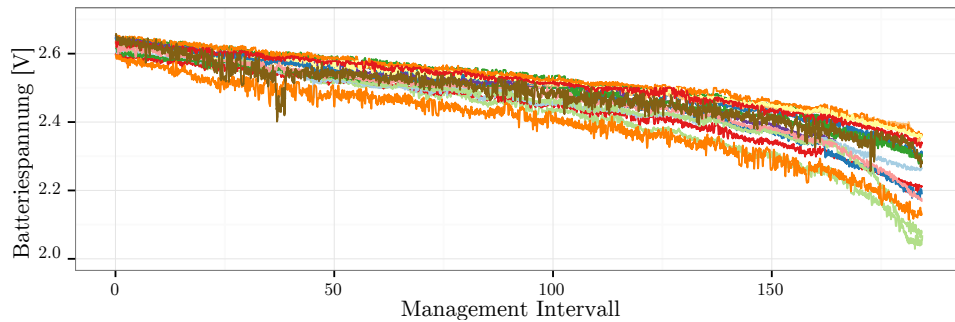


Abbildung 6.14: Batteriespannung der Knoten

eine kleine Ladung an, die die Batterien mit rund 40mAh zusätzlich lud. Weitere Schwankungen können vom Verbrauchsmodell und von Schwankungen zwischen den Knoten verursacht worden sein.

In diesem Messszenario ermöglichte das Energiemanagement den Knoten, von der zusätzlich zur Verfügung stehenden Energie zu profitieren und die Datenqualität zu erhöhen. Dies zeigt, dass das vorgestellte dynamische Energiemanagement mit seinen Budgets in der Lage ist, den Energieverbrauch von Sensorknoten auf der Grundlage von Informationen über Energieverfügbarkeit und Verbrauch zu steuern. So können Lebenszeitziele erreicht und potenziell die Datenqualität erhöht werden.

### 6.3 Zusammenfassung

In zwei Szenarien wurde beispielhaft dargelegt, wie sich Energiebudgets verwenden lassen, um Anwendungen eine Reaktion auf die Veränderung der Energieverfügbarkeit zu ermöglichen. Die Ergebnisse der Versuche zeigen, dass das vorgestellte Energiemanagementkonzept das Erreichen von Lebenszeitzielen im Feld ermöglicht. Mithilfe der Batterieüberwachung konnte sowohl auf den Einfluss der Temperatur auf die Batteriekapazität reagiert als auch die sonst ungenutzte Energie zur Verbesserung der Anwendungsqualität durch Erhöhung des Messintervalls verwendet werden.



## Kapitel 7

# Zusammenfassung und Ausblick

In dieser Arbeit wurde ein Energiemanagementkonzept auf der Basis von Energiebudgets für drahtlose eingebettete Systeme vorgestellt, das es ermöglicht, Lebenszeitziele zu erreichen. Das Konzept berücksichtigt dabei Varianzen, welche in der Energieversorgung und dem Verbrauch auftreten, auf allen Ebenen des Managements. Dies machte eine breite Betrachtung des Themas und die Identifikation von Ursachen auftretender Varianzen notwendig. Weiterhin wurden existierende Ansätze für die Erfassung der verfügbaren Energie, des Verbrauchs und für Managementstrategien eingehend analysiert. Auf Basis dieser Analysen wurden Schwachpunkte identifiziert und Lösungsansätze für alle drei Teilbereiche sowohl konzipiert als auch praktisch umgesetzt und individuell evaluiert. Zudem wurde die Wirksamkeit des Gesamtkonzeptes durch Anwendungsfälle im Bereich der drahtlosen Sensornetze gezeigt.

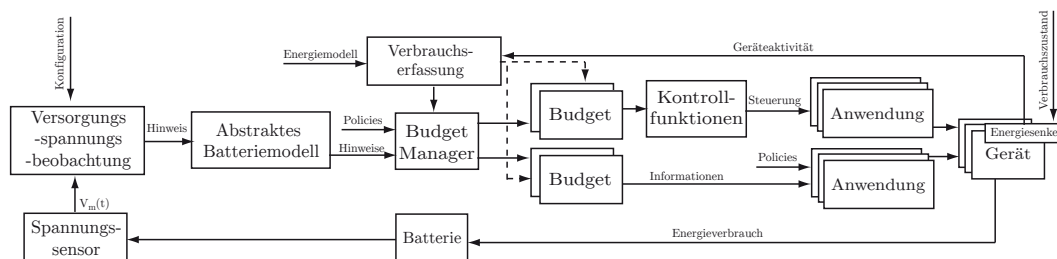


Abbildung 7.1: Entwickelter Energiemanagementansatz

Abbildung 7.1 stellt die Teile des entwickelten Ansatzes und ihren Zusammenhang dar. Im Folgenden werden die einzelnen Teile des Managements gesondert zusammengefasst. Anschließend folgt ein Ausblick, in welche Richtungen das Konzept weiterentwickelt werden könnte.

## 7.1 Zusammenfassung

Bereits in der Einleitung wurden drei Fragestellungen identifiziert, die für einen dynamischen Energiemanagementansatz beantwortet werden mussten:

1. Wie viel Energie steht zur Verfügung?
2. Wie hoch ist der Energieverbrauch?
3. Wie wird die zur Verfügung stehende Energie unter den Verbrauchern aufgeteilt, sodass die Anforderungen der Anwendung erfüllt werden?

Die Beantwortung der ersten Frage machte eine Betrachtung von Batterien und deren Ladezustandserfassung notwendig. Existierende Ansätze leiden unter Hardwarebeziehungsweise Softwareaufwand und Komplexität und/oder mangelnder Flexibilität. Um dem zu begegnen, wurde ein Ansatz entwickelt, der auf eine komplexe Berechnung des Ladezustands verzichtet. Stattdessen wird der Spannungsabfall von Batterien genutzt, um dem Energiemanagement durch Hinweissignale eine Steuerung des Verbrauchs im Hinblick auf das Lebenszeitziel zu ermöglichen. Ein abstraktes Batteriemodell modifiziert die Hinweissignale gegebenenfalls mittels einfacher Regeln, um diese besser an das Verhalten der Batterien anzupassen. Die Wirksamkeit und Eignung des Ansatzes zum Erreichen von Lebenszeitzielen wurde in verschiedenen Szenarien evaluiert. Dabei zeigte sich insbesondere der Einfluss der Temperatur auf die Spannung und Leistungsfähigkeit von Batterien. Dieser konnte durch den Ansatz indirekt erkannt und durch ein Gegensteuern ausgeglichen werden.

Der Beantwortung der zweiten Frage diene die Auseinandersetzung mit den Arten des Energieverbrauchs drahtloser tief eingebetteter Systeme und deren potenzieller Observierbarkeit. Existierende Ansätze, die Hardware zur Erfassung des Verbrauchs nutzen, sind ohne komplexe Berechnungen nicht in der Lage, den Verbrauch feingranular auf einzelne Geräte und deren Modi aufzuschlüsseln. Dies ist nur durch einen Software-basierten Ansatz möglich, wie er in dieser Arbeit generisch und unabhängig von der Anwendung umgesetzt wurde. Da dessen Genauigkeit jedoch vom zugrundeliegenden Verbrauchsmodell abhängt, wurden auf der Grundlage von Messungen realistische Modelle erstellt. Bisherige Ansätze sind nicht in der Lage, sich dynamisch an einen veränderlichen Verbrauch anzupassen. Mit dem entwickelten Ansatz wurden deshalb Mechanismen vorgestellt, die eine Anpassung an Spannungsveränderungen und die variable Effizienz eines Spannungsreglers ermöglichen. Die Funktionsfähigkeit und Genauigkeit des Ansatzes wurden in verschiedenen Szenarien gezeigt.

Die dritte Frage musste durch Betrachtung von Ansätzen zur Energieverwaltung beantwortet werden. Existierende Ansätze steuern den Verbrauch entweder direkt durch Anpassen von Anwendungsparametern oder indirekt mithilfe einer beschränkten Ressource, um die die Anwendungsteile konkurrieren. Wie die Literaturanalyse zeigt, sind viele Ansätze nicht ausreichend in der Lage, auf kurz- und langfristige Bedarfsänderungen zu reagieren. In dieser Arbeit wurde aufgrund der größeren Flexibilität ein indirekter Ansatz umgesetzt und das Konzept der Energiebudgets vorgestellt. Diese setzen das wesentliche Paradigma der Isolation von Anwendungen um, ermöglichen aber zusätzlich eine Kooperation, die Situationen behandelt, in denen die zugeteilte Energie nicht ausreicht oder ungenutzt bleibt. Es wurden Heuristiken zur Energieverteilung und verschiedene Kooperationsmechanismen vorgestellt und deren Eignung mithilfe verschiedener Szenarien evaluiert und bewertet.

## 7.2 Ausblick

Der vorgestellte dynamische Energiemanagementansatz kann auf jeder der drei Ebenen weiterentwickelt werden. Verschiedene Maßnahmen können die Genauigkeit der Abschätzungen von verfügbarer und verbrauchter Energie verbessern. So bietet es sich an, die Temperatur, die einen grossen Einfluss auf die verfügbare Energiemenge in Batterien hat, nicht nur indirekt über den Spannungsabfall einzubeziehen, sondern direkt einfließen zu lassen. Zusätzlich könnte die Temperatur als weiterer Parameter in die Erfassung des Verbrauchs einbezogen werden.

Je nach verwendeter Technologie haben Batterien eine hohe Selbstentladung, die anwendungsabhängig einen signifikanten Anteil an der verbrauchten Energie haben kann. Die durch die Selbstentladung entstehende Last ist dabei von der verbleibenden Ladung in der Batterie und der Temperatur abhängig. Durch eine Einbeziehung dieser Last ließe sich die Qualität der Abschätzung erhöhen.

Da der vorgestellte Ansatz zur Batterieüberwachung selbst keine Aussage über den tatsächlichen Ladestand trifft, diese Information aber gegebenenfalls für Anwendungen von Interesse sein kann, wäre eine Berechnung mithilfe der aus dem Management verfügbaren Daten möglich.

Ist auf der Plattform eine Hardwarelösung zur Erfassung des Ladestandes der Batterie beziehungsweise des Verbrauchs vorhanden, könnten diese Informationen helfen, die Genauigkeit zu erhöhen. Während Ersteres nur eine Anpassung des gesamten Verbrauchs zulässt, könnte mit feingranularen Informationen das Verbrauchsmodell während der Laufzeit verbessert werden, was eine faire Zuordnung der Kosten ermöglicht. Dies setzt jedoch eine Möglichkeit der Zuordnung des erfassten Verbrauchs zu einzelnen Verbrauchern voraus.

Die vorgestellten Energiebudgets sind ein idealer Ausgangspunkt, um in Sensornetzen ein globales Energiemanagement umzusetzen. Sie stellen Informationen über den Energiebedarf, die Verfügbarkeit und den Verbrauch bereit. Durch Verbreitung dieser Informationen innerhalb der Nachbarschaft könnten, analog zu globalen Managementansätzen, Anpassungen des Verhaltens eines Netzes nicht mehr nur indirekt, sondern direkt durch Regeln und Algorithmen durchgeführt werden. Da dies jedoch einen erheblichen Kommunikationsaufwand mit sich bringen kann, ist eine Kosten-Nutzen-Analyse unabdingbar.

Durch die Anpassungsfähigkeit an unterschiedliche Energieverfügbarkeiten bietet sich der vorgestellte Ansatz für die Nutzung in Harvesting-Szenarien an. Abhängig von der Anwendung lässt sich in Phasen mit geringer Energieausbeute der Verbrauch durch entsprechende Energieverteilung so limitieren, dass unter Zuhilfenahme von Prognosen ein kontinuierlicher Betrieb möglich wird.

Die komplexe Dimensionierung der Bedarfswerte der Budgets kann durch Integration von Lernphasen vereinfacht werden. Durch Intervalle, in denen jeweils die minimale und maximale Anwendungsqualität forciert wird, lassen sich Grenzwerte ermitteln. Da sich, wie in dieser Arbeit gezeigt, die Energiekosten im Laufe der Zeit jedoch verändern können, ist eine Wiederholung dieser Lernphasen sinnvoll.

Die Nutzung des Ansatzes lässt sich durch Integration in einen modellgetriebenen Entwicklungsprozess vereinfachen. Analog zur Echtzeitanalyse kann eine Energieanalyse bekannte Kosten von Softwarekomponenten und der Geräte mit den Anforderungen des Nutzers hinsichtlich gewünschter Anwendungsfrequenzen kombinieren. Teilt der Nutzer die verfügbare Energie auf einzelne Anwendungsteile auf, lässt sich die potenzielle Laufzeit errechnen. Durch Angabe des Lebenszeitziels kann anschließend eine Analyse und Optimierung der Parameter erfolgen. Bei der in [63] skizzierten Werkzeugunterstützung und Generierung von strukturbeschreibendem Quellcode aus der Modellebene können die Energiesenken der Verbrauchserfassung entsprechend den Anforderungen an die Granularität konfiguriert und Budgets mit ihren Parametern angelegt sowie mit den Anwendungsteilen verbunden werden.

Anhang A

Abbildungen

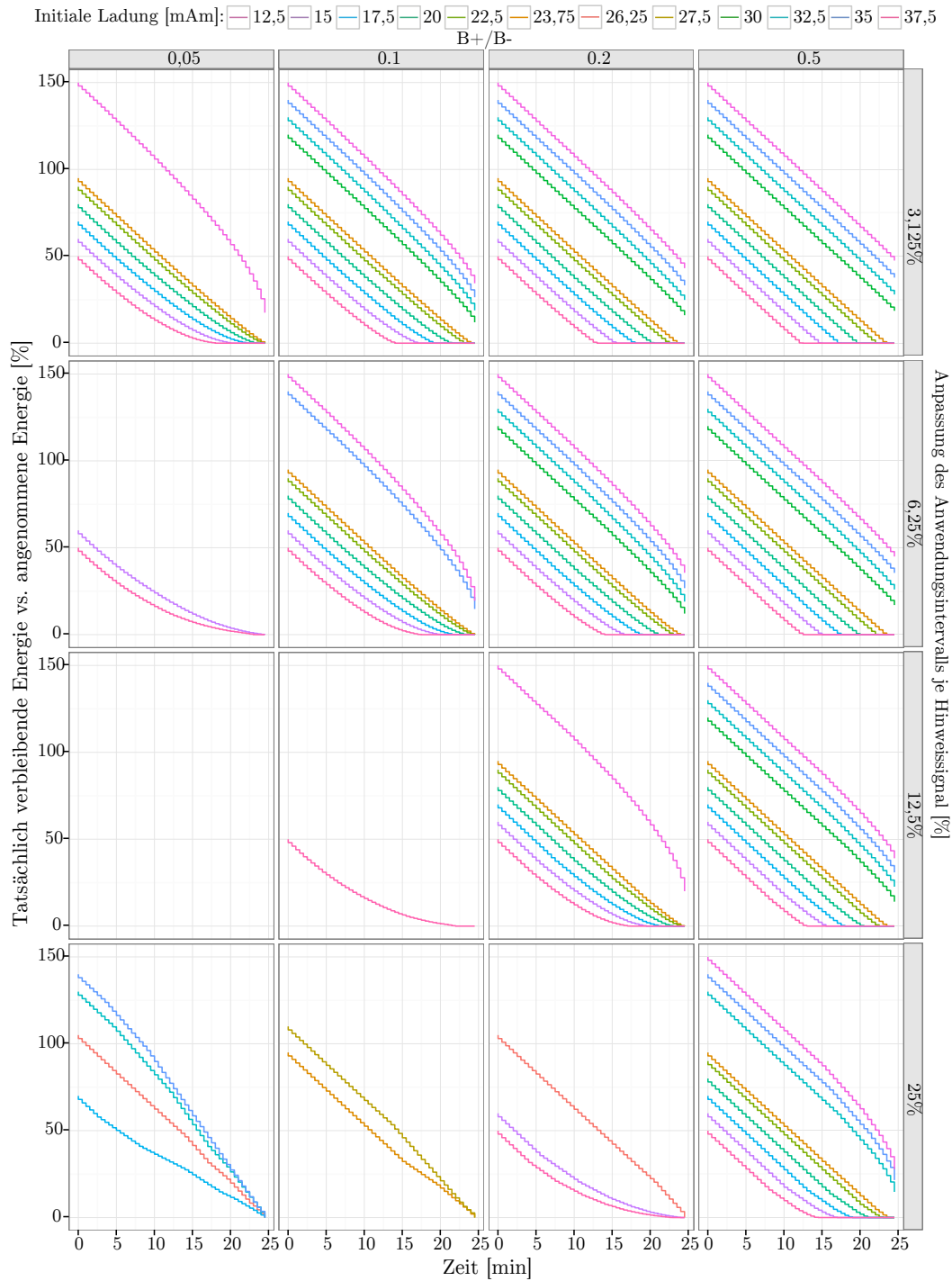


Abbildung A.1: Verlauf der verbleibenden Energie bei verschiedenen Parameterkombinationen und initialen Ladungen: Es werden nur Fälle dargestellt, in denen das Lebenszeitziel nicht erreicht wurde oder mehr als 10% der Energie ungenutzt blieb.

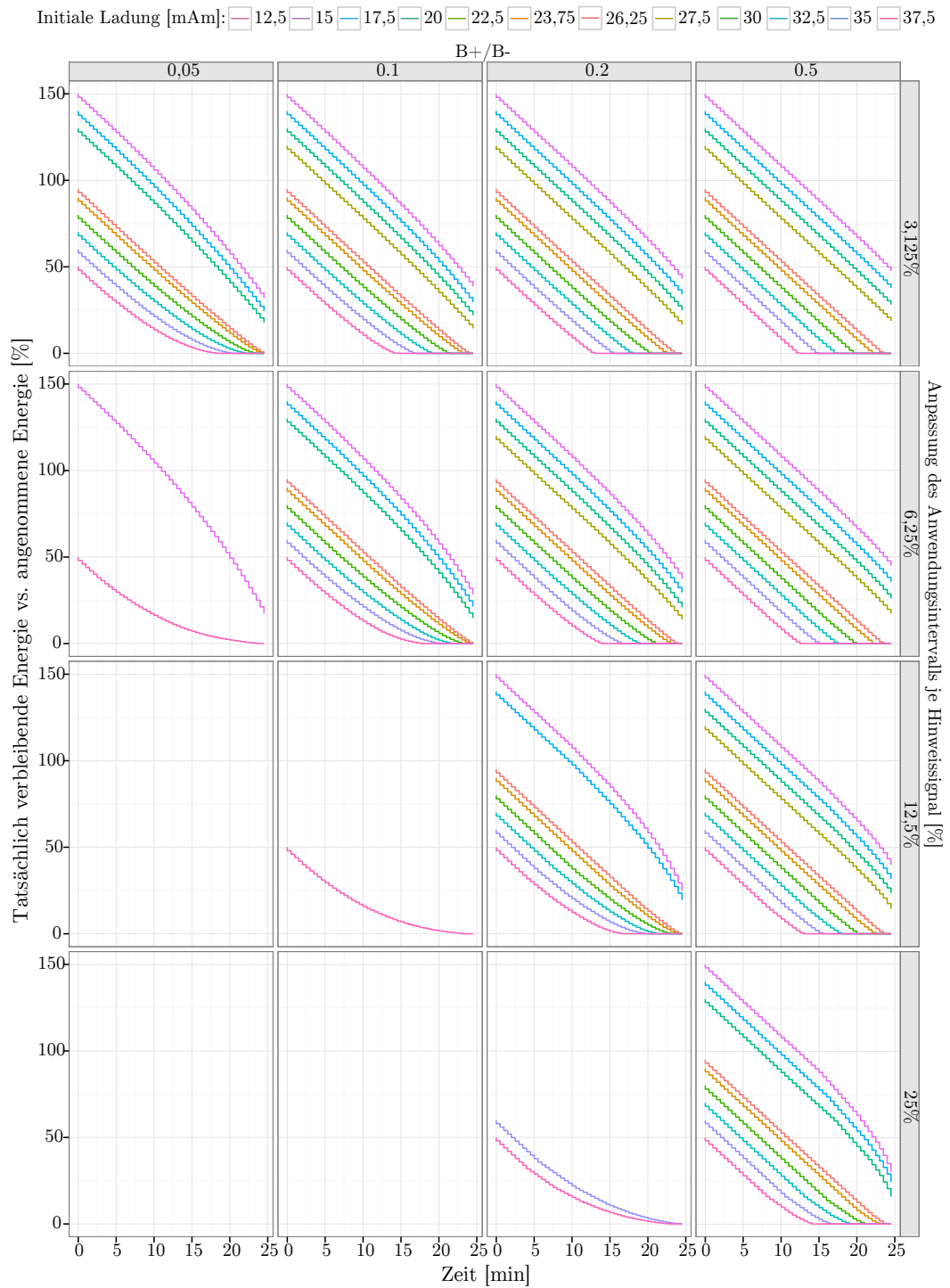


Abbildung A.2: Verlauf der verbleibenden Energie aller nicht erfolgreichen Szenarien unter verschiedenen Parameterkombinationen und initialen Ladungen: Die kleinstmögliche obere Schranke wurde auf den Spannungsabfall eines Intervalls limitiert.





Anhang B

Tabellen

Tabelle B.1: Auswertung der gemessenen Stromaufnahme, aufgeteilt nach Betriebsmodi für 2,2V und 3,0V

	3,0V [mA]					2,2V [mA]				
	Min	Max	Median	Streuung	Datenblatt	Min	Max	Median	Streuung	
Ip <sub>m3P</sub> PL3	0,0026	0,1320	0,0028	5074%	0,0022		nicht möglich			
Ip <sub>m3P</sub> PL2	0,0025	0,1314	0,0028	5093%	0,0022	0,0021	0,0329	0,00231	1447%	
Ip <sub>m3P</sub> PL1	0,0024	0,1307	0,0027	5277%	0,0021	0,0020	0,0332	0,00221	1533%	
Ip <sub>m3P</sub> PL0	0,0023	0,1294	0,0026	5407%	0,0020	0,0019	0,0336	0,00211	1620%	
1MHzPL3	0,2729	0,4164	0,2852	53%	0,28		nicht möglich			
1MHzPL2	0,259	0,397	0,2703	54%	0,27	0,2514	0,2912	0,2640	16%	
1MHzPL1	0,2374	0,3764	0,2487	59%	0,25	0,2323	0,2698	0,2445	16%	
1MHzPL0	0,2132	0,3361	0,2235	58%	0,23	0,2086	0,2416	0,2199	16%	
8MHzPL3	1,808	1,987	1,870	10%	1,85		nicht möglich			
8MHzPL2	1,695	1,875	1,755	11%	1,75	1,679	1,782	1,737	6%	
8MHzPL1	1,531	1,711	1,587	12%	1,55	1,513	1,615	1,572	7%	
8MHzPL0	1,342	1,533	1,394	14%	1,35	1,325	1,419	1,380	7%	
16MHzPL3	3,450	3,692	3,597	7%	3,65		nicht möglich			
16MHzPL2	3,228	3,477	3,369	8%	3,45	3,205	3,418	3,349	7%	
20MHzPL3	4,292	4,604	4,495	7%	4,55		nicht möglich			
RF tx +10dBm	26,64	38,4	34,44	44%	33,0022	26,63	38,17	34,42	43%	
RF tx +0dBm	17,05	18,34	17,70	8%	17,8022	16,89	18,17	17,54	8%	
RF tx -6dBm	17,09	17,95	17,48	5%	17,8022	16,54	17,694	17,39	5%	
RF rx current	15,82	16,55	16,22	5%	16,0022	15,64	16,50	16,06	5%	
RF rx sensitive	17,40	18,20	17,85	5%	17,9022	17,23	18,13	17,69	5%	
RF intermediate	8,947	9,501	9,19	6%	9,5022	8,798	9,335	9,05	6%	
RF idle	1,599	1,765	1,657	10%	1,7022	1,541	1,643	1,599	7%	
adc ref	0,121	0,239	0,126	98%	0,1522	0,112	0,149	0,116	33%	
	0,095	0,216	0,107	124%	0,1022	0,093	0,130	0,105	40%	

Tabelle B.2: Pakete pro Sekunde mit und ohne Verbrauchserfassung im Ping-Pong-Verfahren: kleine Pakete mit 16 Byte Nutzdaten, große Pakete mit 60 Byte

	kleine Pakete pro Sekunde			große Pakete pro Sekunde		
	Ohne	Pessimistisch	Optimistisch	Ohne	Pessimistisch	Optimistisch
1 MHz	38,6	35,2	36,8	16,3	15,6	15,9
4 MHz	81,6	77,5	79,5	35,5	34,7	35,1
8 MHz	100,7	97,6	99,1	44,5	43,9	44,2
16 MHz	114,1	112,1	113,1	51,0	50,6	50,8
20 MHz	117,3	115,6	116,4	52,5	52,2	52,3

Tabelle B.3: Pakete pro Sekunde mit und ohne Verbrauchserfassung im Ping-Pong-Verfahren bei Einbeziehung eines Spannungsreglers: kleine Pakete mit 16 Byte Nutzdaten, große Pakete mit 60 Byte

	kleine Pakete pro Sekunde		große Pakete pro Sekunde	
	vollständige Abrechnung	reine Effizienzberechnung	vollständige Abrechnung	reine Effizienzberechnung
1 MHz	32,1	36,0	14,95	15,8
4 MHz	73,8	78,6	33,95	34,95
8 MHz	94,6	98,4	43,3	44,1
16 MHz	110,1	112,7	50,2	50,7
20 MHz	113,9	116,1	51,8	52,25

Tabelle B.4: Pakete pro Sekunde im Ping-Pong-Verfahren mit und ohne Energiemanagement mit unterschiedlichen Zuordnungsgebenen, jedoch immer mit Verbrauchserfassung: kleine Pakete mit 16 Byte Nutzdaten, große Pakete mit 60 Byte

Mhz	kleine Pakete pro Sekunde				große Pakete pro Sekunde			
	Ohne	Gerätebasiert	Anfragebasiert	Pfadbasiert	Ohne	Gerätebasiert	Anfragebasiert	Pfadbasiert
1 Mhz	36,8	36,2	30,05	28,35	15,9	15,8	14,5	14,1
4 Mhz	79,5	78,8	70,8	68,5	35,1	35,0	33,3	32,8
8 Mhz	99,1	98,6	92,05	90,1	44,2	44,1	42,7	42,3
16 Mhz	113,1	112,75	108,35	107,0	50,8	50,7	49,8	49,5
20 Mhz	116,4	116,1	112,35	111,15	52,3	52,25	51,5	51,2



# Literaturverzeichnis

- [1] KAHN, J. M. ; KATZ, R. H. ; PISTER, K. S. J.: Next Century Challenges: Mobile Networking for "Smart Dust". In: *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*. Seattle, USA, 1999, S. 271–278
- [2] ESTRIN, Deborah ; GIROD, Lewis ; POTTIE, Greg ; SRIVASTAVA, Mani: Instrumenting the world with wireless sensor networks. In: *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*. Salt Lake City, USA, 2001, S. 2033–2036
- [3] VAN DAM, Karen ; PITCHERS, Steve ; BARNARD, Mike: Body area networks: Towards a wearable future. In: *Proceedings of the Wireless World Research Forum kick off meeting*. München, Deutschland, 2001, S. 6–7
- [4] ASHTON, Kevin: That 'Internet of Things' Thing. In: *RFiD Journal* 22 (2009), S. 97–114
- [5] WERNER-ALLEN, Geoffrey ; LORINCZ, Konrad ; RUIZ, Mario ; MARCILLO, Omar ; JOHNSON, Jeff ; LEES, Jonathan ; WELSH, Matt: Deploying a wireless sensor network on an active volcano. In: *IEEE Internet Computing* 10 (2006), Nr. 2, S. 18–25
- [6] MARTINEZ, Kirk ; ONG, Royan ; HART, Jane: Glacsweb: a sensor network for hostile environments. In: *Proceedings of the First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*. Santa Clara, USA, 2004, S. 81–87
- [7] BURRELL, Jenna ; BROOKE, Tim ; BECKWITH, Richard: Vineyard computing: Sensor networks in agricultural production. In: *IEEE Pervasive Computing* 3 (2004), Nr. 1, S. 38–45
- [8] LANGENDOEN, Koen ; BAGGIO, Aline ; VISSER, Otto: Murphy loves potatoes: Experiences from a pilot sensor network deployment in precision agriculture. In: *Proceedings of the 20th International Conference on Parallel and Distributed Processing*. Rhodos, Griechenland, 2006, S. 174–174

- [9] JUANG, Philo ; OKI, Hidekazu ; WANG, Yong ; MARTONOSI, Margaret ; PEH, Li S. ; RUBENSTEIN, Daniel: Energy-efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. In: *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems*. San Jose, USA, 2002, S. 96–107
  
- [10] MAINWARING, Alan ; CULLER, David ; POLASTRE, Joseph ; SZEWCZYK, Robert ; ANDERSON, John: Wireless sensor networks for habitat monitoring. In: *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. Atlanta, USA, 2002, S. 88–97
  
- [11] KIM, Sukun ; PAKZAD, Shamim ; CULLER, David ; DEMMEL, James ; FENVES, Gregory ; GLASER, Steven ; TURON, Martin: Health monitoring of civil infrastructures using wireless sensor networks. In: *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*. Cambridge, USA, 2007, S. 254–263
  
- [12] CERIOTTI, Matteo ; CORRÀ, Michele ; D’ORAZIO, Leandro ; DORIGUZZI, Roberto ; FACCHIN, Daniele ; GUNA, Stefan ; JESI, Gian P. ; CIGNO, Renato L. ; MOTTOLA, Luca ; MURPHY, Amy L. ; PESCALLI, Massimo ; PICCO, Gian P. ; PREGNOLATO, Denis ; TORGHELE, Carloalberto: Is there light at the ends of the tunnel? Wireless sensor networks for adaptive lighting in road tunnels. In: *Proceedings of the 10th International Conference on Information Processing in Sensor Networks*. Chicago, USA, 2011, S. 187–198
  
- [13] JEDERMANN, Reiner ; BEHRENS, Christian ; WESTPHAL, Detmar ; LANG, Walter: Applying autonomous sensor systems in logistics—Combining sensor networks, RFIDs and software agents. In: *Sensors and Actuators A: Physical* 132 (2006), Nr. 1, S. 370–375
  
- [14] RIEM-VIS, Ruud: Cold chain management using an ultra low power wireless sensor network. In: *Proceedings of the MobiSys 2004 Workshop on Applications of Mobile Embedded Systems*. Boston, USA, 2004, S. 21–23
  
- [15] SIMON, Gyula ; MARÓTI, Miklós ; LÉDECZI, Ákos ; BALOGH, György ; KUSY, Branislav ; NÁDAS, András ; PAP, Gábor ; SALLAI, János ; FRAMPTON, Ken: Sensor network-based countersniper system. In: *Proceedings of the 2nd international conference on Embedded networked sensor systems*. Baltimore, USA, 2004, S. 1–12

- [16] HE, Tian ; KRISHNAMURTHY, Sudha ; LUO, Liqian ; YAN, Ting ; GU, Lin ; STOLERU, Radu ; ZHOU, Gang ; CAO, Qing ; VICAIRE, Pascal ; STANKOVIC, John A. u. a.: VigilNet: An integrated sensor network system for energy-efficient surveillance. In: *ACM Transactions on Sensor Networks* 2 (2006), Nr. 1, S. 1–38
- [17] DRESSLER, Falko ; DIETRICH, Isabel: Lifetime Analysis in Heterogeneous Sensor Networks. In: *Proceedings of the 9th EUROMICRO Conference on Digital System Design - Architectures, Methods and Tools*. Dubrovnik, Kroatien, 2006, S. 606–613
- [18] LATRÉ, Benoît ; BRAEM, Bart ; MOERMAN, Ingrid ; BLONDIA, Chris ; DE-MEESTER, Piet: A survey on wireless body area networks. In: *Wireless Networks* 17 (2011), Nr. 1, S. 1–18
- [19] MADDEN, Samuel R. ; FRANKLIN, Michael J. ; HELLERSTEIN, Joseph M. ; HONG, Wei: TinyDB: An acquisitional query processing system for sensor networks. In: *ACM Transactions on Database Systems* 30 (2005), Nr. 1, S. 122–173
- [20] REDDY, Thomas: *Linden's Handbook of Batteries, 4th Edition*. Mcgraw-hill, 2010. – ISBN 9780071624190
- [21] LACHENMANN, Andreas ; MARRÓN, Pedro J. ; MINDER, Daniel ; ROTHERMEL, Kurt: Meeting lifetime goals with energy levels. In: *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*. Sydney, Australien, 2007, S. 131–144
- [22] KANSAL, Aman ; SRIVASTAVA, Mani B.: An Environmental Energy Harvesting Framework for Sensor Networks. In: *Proceedings of the 2003 International Symposium on Low Power Electronics and Design*. Seoul, Korea, 2003, S. 481–486
- [23] DECKER, Christian: *Prozessorganisation in eingebetteten, ubiquitären Rechnersystemen*. Karlsruhe, Karlsruhe Institute of Technology, Dissertation, 2009
- [24] BANGA, Gaurav ; DRUSCHEL, Peter ; MOGUL, Jeffrey C.: Resource containers: A new facility for resource management in server systems. In: *Proceedings of the Third Symposium on Operating Systems Design and Implementation*. New Orleans, USA, 1999, S. 45–58
- [25] NASS, Tobias ; ANDERSEN, Audun: *Powering Low-Power RF Products*. <http://www.ti.com/lit/an/swra173b/swra173b.pdf>, Design Note DB019, Texas Instruments, [online] letzter Aufruf: 28.02.2016

- [26] PANASONIC: *Datenblatt eneloop BK-3MCC AA Nickel-Metal Hydride Battery*. <http://www.eneloop.eu/>, [online] letzter Aufruf: 27.02.2016
- [27] NIEMINEN, Lauri: *Disposable battery capacity tests*. <http://www.laurinieminen.com/>, [online] letzter Aufruf: 27.02.2016
- [28] CHEN, Min ; RINCON-MORA, Gabriel A.: Accurate electrical battery model capable of predicting runtime and IV performance. In: *IEEE Transactions on Energy Conversion* 21 (2006), Nr. 2, S. 504–511
- [29] WENZL, Heinz: *Batterietechnik: Optimierung der Anwendung-Betriebsführung-Systemintegration*. expert-Verlag, 1999. – ISBN 9783816916918
- [30] SZENTE-VARGA, D. ; HORVATH, D. ; RENCZ, M.: Ni-MH battery modelling for ambient intelligence applications. In: *Proceedings of the Symposium on Design, Test, Integration and Packaging of MEMS/MOEMS*. Stresa lago Maggiore, Italien, 2007, S. 332–337
- [31] RAO, Venkat ; SINGHAL, Gaurav ; KUMAR, Anshul ; NAVET, Nicolas: Battery model for embedded systems. In: *Proceedings of the 18th International Conference on VLSI Design*. Kolkata, Indien, 2005, S. 105–110
- [32] FURSET, Kjartan ; HOFFMAN, Peter: *High pulse drain impact on CR2032 coin cell battery capacity*. <http://m.eet.com/media/1121454/c0924post.pdf>, White Paper, Nordic Semiconductor/Energizer, [online] letzter Aufruf: 28.02.2016
- [33] JENSEN, Mathias: *Coin Cells And Peak Current Draw*. <http://www.ti.com/lit/wp/swra349/swra349.pdf>, White Paper SWRA349, Texas Instruments, [online] letzter Aufruf: 28.02.2016
- [34] PARK, Chulsung ; LAHIRI, Kanishka ; RAGHUNATHAN, Anand: Battery discharge characteristics of wireless sensor nodes: An experimental analysis. In: *Proceedings of the Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*. Santa Clara, USA, 2005, S. 430–440
- [35] PADMANABH, Kumar ; ROY, Rajarshi: Maximum lifetime routing in wireless sensor network by minimizing rate capacity effect. In: *Proceedings of the International Conference on Parallel Processing Workshops*. Columbus, USA, 2006, S. 165–174
- [36] CHIASSERINI, Carla-Fabiana ; RAO, Ramesh R.: Routing protocols to maximize battery efficiency. In: *Proceedings of the 21st Century Military Communications Conference*. Los Angeles, USA, 2000, S. 496–500



- [37] CHAU, Chi-Kin ; QIN, Fei ; SAYED, Samir ; WAHAB, Muhammad H. ; YANG, Yang: Harnessing battery recovery effect in wireless sensor networks: Experiments and analysis. In: *IEEE Journal on Selected Areas in Communications* 28 (2010), Nr. 7, S. 1222–1232
- [38] BUCHMAN, Isidor: *Batteries in a Portable World: A Handbook on Rechargeable Batteries for Non-Engineers*. Cadex Electronics Inc Richmond, 2001. – ISBN 978–0968211823
- [39] WANG, Wensi S. ; O'DONNELL, Terence ; WANG, Ningning ; HAYES, Michael ; O'FLYNN, Brendan ; O'MATHUNA, C: Design considerations of sub-mW indoor light energy harvesting for wireless sensor systems. In: *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 6 (2010), Nr. 2, S. 6
- [40] STOJCEV, Mile K. ; KOSANOVIC, Mirko R. ; GOLUBOVIC, Ljubiša R: Power management and energy harvesting techniques for wireless sensor nodes. In: *Proceedings of the 9th International Conference on Telecommunication in Modern Satellite, Cable, and Broadcasting Services*. Niš, Serbien, 2009, S. 65–72
- [41] RAJU, Murugavel ; GRAZIER, Mark: *Energy Harvesting: ULP Meets Energy Harvesting: A Game-Changing Combination for Design Engineers*. <http://www.ti.com/lit/pdf/slyy018>, White Paper, Texas Instruments, [online] letzter Aufruf: 28.02.2016
- [42] PARADISO, Joseph A. ; STARNER, Thad: Energy scavenging for mobile and wireless electronics. In: *IEEE Pervasive Computing* 4 (2005), Nr. 1, S. 18–27
- [43] PARADISO, Joseph A. ; FELDMEIER, Mark: A Compact, Wireless, Self-Powered Pushbutton Controller. In: *Proceedings of the 3rd International Conference on Ubiquitous Computing*. Atlanta, USA, 2001, S. 299–304
- [44] SUDEVALAYAM, Sujesha ; KULKARNI, Purushottam: Energy harvesting sensor nodes: Survey and implications. In: *IEEE Communications Surveys & Tutorials* 13 (2011), Nr. 3, S. 443–461
- [45] RAGHUNATHAN, Vijay ; KANSAL, Aman ; HSU, Jason ; FRIEDMAN, Jonathan ; SRIVASTAVA, Mani: Design Considerations for Solar Energy Harvesting Wireless Embedded Systems. In: *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*. Los Angeles, USA, 2005, S. 457–462
- [46] SIMJEE, Farhan ; SHARMA, Devyani ; CHOU, Pai H.: Everlast: Long-life, Supercapacitor-operated Wireless Sensor Node. In: *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*. San Diego, USA, 2005, S. 315–315

- [47] JIANG, Xiaofan ; POLASTRE, Joseph ; CULLER, David: Perpetual Environmentally Powered Sensor Networks. In: *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*. Los Angeles, USA, 2005, S. 463–468
- [48] KANSAL, Aman ; HSU, Jason ; ZAHEDI, Sadaf ; SRIVASTAVA, Mani B.: Power management in energy harvesting sensor networks. In: *ACM Transactions on Embedded Computing Systems (TECS)* 6 (2007), Nr. 4, S. 32
- [49] PIORNO, J R. ; BERGONZINI, Carlo ; ATIENZA, David ; ROSING, T S.: Prediction and management in energy harvested wireless sensor nodes. In: *Proceedings of the 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology*. Aalborg, Dänemark, 2009, S. 6–10
- [50] PARK, Sung ; SAVVIDES, Andreas ; SRIVASTAVA, Mani: Battery capacity measurement and analysis using lithium coin cell battery. In: *Proceedings of the 2001 international symposium on Low power electronics and design*. Huntington Beach, USA, 2001, S. 382–387
- [51] SZEWCZYK, Robert ; POLASTRE, Joseph ; MAINWARING, Alan ; CULLER, David: Lessons from a sensor network expedition. In: *Proceedings of the 1th European Conference on Wireless Sensor Networks*. Berlin, Deutschland, 2004, S. 307–322
- [52] ON SEMICONDUCTOR: *Datenblatt NCP1400A*. <http://onsemi.com>, [online] letzter Aufruf: 28.02.2016
- [53] ATMEL: *Datenblatt 8/16-bit Atmel XMEGA D4 Microcontroller*. <http://atmel.com>, [online] letzter Aufruf: 28.02.2016
- [54] TEXAS INSTRUMENTS: *Datenblatt CC430F61xx 16-Bit Ultra-Low-Power MCU*. <http://www.ti.com>, [online] letzter Aufruf: 27.02.2016
- [55] ENERGY MICRO: *Datenblatt EFM32G890*. <http://energymicro.com>, [online] letzter Aufruf: 12.03.2014
- [56] STMICROELECTRONICS: *Datenblatt STM32F205xx STM32F207xx*. <http://st.com>, [online] letzter Aufruf: 28.02.2016
- [57] REAL TIME ENGINEERS LTD.: *FreeRTOS*. <http://www.freertos.org>, [online] letzter Aufruf: 28.02.2016

- [58] BHATTI, Shah ; CARLSON, James ; DAI, Hui ; DENG, Jing ; ROSE, Jeff ; SHETH, Anmol ; SHUCKER, Brian ; GRUENWALD, Charles ; TORGERSON, Adam ; HAN, Richard: MANTIS OS: An embedded multithreaded operating system for wireless micro sensor platforms. In: *Mobile Networks and Applications* 10 (2005), Nr. 4, S. 563–579
- [59] HAN, Chih-Chieh ; RENGASWAMY, Ram K. ; SHEA, Roy ; KOHLER, Eddie ; SRIVASTAVA, Mani: A dynamic operating system for sensor networks. In: *Third International Conference on Mobile Systems, Applications, And Services*. Seattle, USA, 2005, S. 163–176
- [60] LEVIS, Philip ; MADDEN, Sam ; POLASTRE, Joseph ; SZEWCZYK, Robert ; WHITEHOUSE, Kamin ; WOO, Alec ; GAY, David ; HILL, Jason ; WELSH, Matt ; BREWER, Eric u. a.: TinyOS: An operating system for sensor networks. In: *Ambient intelligence*. Springer Berlin Heidelberg, 2005. – ISBN 978–3–540–27139–0, S. 115–148
- [61] DUNKELS, Adam ; GRONVALL, Bjorn ; VOIGT, Thiemo: Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors. In: *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*. Tampa, USA, 2004, S. 455–462
- [62] WALTHER, Karsten ; NOLTE, Jörg: A flexible scheduling framework for deeply embedded systems. In: *Proceedings of the 4th IEEE International Symposium on Embedded Computing*. Niagara Falls, Kanada, 2007, S. 784–791
- [63] WALTHER, Karsten: *Ein ereignisbasiertes Betriebssystemkonzept für tief eingebettete Steuersysteme*. Cottbus, BTU Cottbus, Dissertation, 2009
- [64] DUNKELS, Adam ; SCHMIDT, Oliver ; VOIGT, Thiemo ; ALI, Muneeb: Protothreads: simplifying event-driven programming of memory-constrained embedded systems. In: *Proceedings of the 4th international conference on Embedded networked sensor systems*. Boulder, USA, 2006, S. 29–42
- [65] MCCARTNEY, William P. ; SRIDHAR, Nigamanth: Abstractions for safe concurrent programming in networked embedded systems. In: *Proceedings of the 4th international conference on Embedded networked sensor systems*. Boulder, USA, 2006, S. 167–180
- [66] SORBER, Jacob ; KOSTADINOV, Alexander ; GARBER, Matthew ; BRENNAN, Matthew ; CORNER, Mark D. ; BERGER, Emery D.: Eon: A Language and Runtime System for Perpetual Systems. In: *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*. Sydney, Australien, 2007, S. 161–174

- [67] WIEDENHOFT, Geovani R. ; WANNER, Lucas F. ; GRACIOLI, Giovanni ; FRÖHLICH, Antônio A.: Power management in the EPOS system. In: *ACM SIGOPS Operating Systems Review* 42 (2008), Nr. 6, S. 71–80
- [68] NEUGEBAUER, Rolf ; MCAULEY, Derek: Energy is just another resource: Energy accounting and energy pricing in the Nemesis OS. In: *Proceedings of the Eighth Workshop on Hot Topics in Operating Systems*. Elmau, Deutschland, 2001, S. 67–72
- [69] ZENG, Heng ; ELLIS, Carla S. ; LEBECK, Alvin R. ; VAHDAT, Amin: ECO-System: Managing energy as a first class operating system resource. In: *Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*. San Jose, USA, 2002, S. 123–132
- [70] KANG, S.M. ; LEBLEBICI, Y.: *CMOS Digital Integrated Circuits Analysis & Design, 3th Edition*. McGraw-Hill Education, 2002 (McGraw-Hill Series in Electrical and Computer Engineering). – ISBN 9780072460537
- [71] GHATTAS, Rony ; DEAN, Alexander G.: Energy management for commodity short-bit-width microcontrollers. In: *Proceedings of the 2005 international conference on Compilers, architectures and synthesis for embedded systems*. San Francisco, USA, 2005, S. 32–42
- [72] CHO, Youngjin ; KIM, Younghyun ; CHANG, Naehyuck: PVS: passive voltage scaling for wireless sensor networks. In: *Proceedings of the 2007 international symposium on Low power electronics and design*. Portland, USA, 2007, S. 135–140
- [73] KIM, Seunghoon ; DICK, Robert P. ; JOSEPH, Russ: Power deregulation: eliminating off-chip voltage regulation circuitry from embedded systems. In: *Proceedings of the 5th IEEE/ACM International Conference on Hardware/Software Codesign and System Synthesis*. Salzburg, Österreich, 2007, S. 105–110
- [74] CHO, Youngjin ; KIM, Younghyun ; JOO, Yongsoo ; LEE, Kyungsoo ; CHANG, Naehyuck: Simultaneous Optimization of Battery-aware Voltage Regulator Scheduling with Dynamic Voltage and Frequency Scaling. In: *Proceedings of the 13th International Symposium on Low Power Electronics and Design*. Bangalore, India, 2008, S. 309–314
- [75] SZEWCZYK, Robert ; LEVIS, Philip ; TURON, Martin ; NACHMAN, Lama ; BUONADONNA, Philip ; HANDZISKI, Vlado: *Tep 112: Microcontroller power management*. <http://www.nordicsemi.com>, [online] letzter Aufruf: 28.02.2016

- [76] KLUES, Kevin ; HANDZISKI, Vlado ; LU, Chenyang ; WOLISZ, Adam ; CULLER, David ; GAY, David ; LEVIS, Philip: Integrating concurrency control and energy management in device drivers. In: *Proceedings of Twenty-first ACM SIGOPS Symposium on Operating Systems Principles*. Stevenson, USA, 2007, S. 251–264
- [77] SIEBER, André ; WALTHER, Karsten ; NÜRNBERGER, Stefan ; NOLTE, Jörg: Implicit sleep mode determination in power management of event-driven deeply embedded systems. In: *Proceedings of the 7th International Conference on Wired/Wireless Internet Communications*. Enschede, Niederlande : Springer, 2009, S. 13–23
- [78] KARNAPKE, Reinhardt: *Unidirectional Links in Wireless Sensor Networks*. Cottbus, BTU Cottbus, Dissertation, 2013
- [79] SCHILLER, Jochen H.: *Mobile communications, 2th Edition*. Pearson Education, 2003. – ISBN 9780321123817
- [80] SCHILLER, Jochen ; LIERS, Achim ; RITTER, Hartmut ; WINTER, Rolf ; VOIGT, Thiemo: ScatterWeb - Low Power Sensor Nodes and Energy Aware Routing. In: *Proceedings of the 38th Hawaii International Conference on System Sciences*. Big Island, USA, 2005, S. 286c
- [81] MIN, Rex ; CHANDRAKASAN, Anantha: MobiCom poster: top five myths about the energy consumption of wireless communication. In: *ACM SIGMOBILE Mobile Computing and Communications Review* 7 (2003), Nr. 1, S. 65–67
- [82] TEXAS INSTRUMENTS: *Datenblatt CC2520 2.4 GHZ IEEE 802.15.4/ZIGBEE® RF TRANSCEIVER*. <http://www.ti.com>, [online] letzter Aufruf: 28.02.2016
- [83] TEXAS INSTRUMENTS: *Datenblatt CC2570, CC2571 1- and 8-channel ANT<sup>TM</sup> RF Network Processors*. <http://www.ti.com>, [online] letzter Aufruf: 28.02.2016
- [84] NORDIC SEMICONDUCTOR: *Datenblatt nRF8001 Single-chip Bluetooth® low energy solution*. <http://www.nordicsemi.com>, [online] letzter Aufruf: 28.02.2016
- [85] TEXAS INSTRUMENTS: *Datenblatt TI SimpleLink<sup>TM</sup> CC3000 Module – Wi-Fi 802.11b/g Network Processor*. <http://www.ti.com>, [online] letzter Aufruf: 28.02.2016
- [86] TEXAS INSTRUMENTS: *Datenblatt CC1101 Low-Power Sub-1 GHz RF Transceiver*. <http://www.ti.com>, [online] letzter Aufruf: 28.02.2016

- [87] BLUETOOTH SIG: *Specification of the Bluetooth System*. <http://www.bluetooth.com>, [online] letzter Aufruf: 27.02.2016
- [88] ROBERTS, Nathan E. ; WENTZLOFF, David D.: A 98nW wake-up radio for wireless body area networks. In: *Proceedings of the IEEE Radio Frequency Integrated Circuits Symposium*. Montreal, Kanada, 2012, S. 373–376
- [89] LE-HUY, Philippe ; ROY, Sébastien: Low-power wake-up radio for wireless sensor networks. In: *Mobile Networks and Applications* 15 (2010), Nr. 2, S. 226–236
- [90] GU, Lin ; STANKOVIC, John A.: Radio-Triggered Wake-Up Capability for Sensor Networks. In: *Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium*. Toronto, Kanada, 2004, S. 27–37
- [91] POLASTRE, Joseph ; HILL, Jason ; CULLER, David: Versatile Low Power Media Access for Wireless Sensor Networks. In: *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*. Baltimore, USA, 2004, S. 95–107
- [92] YE, Wei ; HEIDEMANN, John ; ESTRIN, Deborah: An energy-efficient MAC protocol for wireless sensor networks. In: *Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies* Bd. 3. New York, USA, 2002, S. 1567–1576
- [93] CHATTERJEA, S ; VAN HOESEL, LFW ; HAVINGA, PJM: AI-LMAC: an adaptive, information-centric and lightweight MAC protocol for wireless sensor networks. In: *Proceedings of the Intelligent Sensors, Sensor Networks and Information Processing Conference*. Melbourne, Australien, 2004, S. 381–388
- [94] HOESEL, Lodewijk F. ; HAVINGA, PJM: A lightweight medium access protocol (LMAC) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches. In: *Proceedings of the 1st International Workshop on Networked Sensing Systems*. Tokio, Japan, 2004, S. 205–208
- [95] LU, Gang ; KRISHNAMACHARI, Bhaskar ; RAGHAVENDRA, Cauligi S.: An adaptive energy-efficient and low-latency MAC for data gathering in sensor networks. In: *Proceedings of the 18th International Parallel and Distributed Processing Symposium*. Santa Fe, USA, 2004, S. 224–236
- [96] SONG, Jianping ; HAN, Song ; MOK, Aloysius K. ; CHEN, Deji ; LUCAS, Mike ; NIXON, Mark: WirelessHART: Applying wireless technology in real-time industrial process control. In: *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium*. St. Louis, USA, 2008, S. 377–386



- [97] DAM, Tijs van ; LANGENDOEN, Koen: An adaptive energy-efficient MAC protocol for wireless sensor networks. In: *Proceedings of the 1st international conference on Embedded networked sensor systems*. Los Angeles, California, USA, 2003, S. 171–180
- [98] MANK, Stephan ; KARNAPKE, Reinhardt ; NOLTE, Jörg: MLMAC-UL and ECTS-MAC - Two MAC Protocols for Wireless Sensor Networks with Unidirectional Links. In: *Proceedings of the Third International Conference on Sensor Technologies and Applications*. Athen, Griechenland, 2009, S. 623–629
- [99] MCKLEROY, Vel S. ; GALBRAITH, Jennifer S. ; CUMMINGS, Beverley ; JONES, Patricia ; HARSHBARGER, Camilla ; COLLINS, Charles ; GELAUE, Deborah ; CAREY, James W. ; ADAPT, Team: Adapting evidence-based behavioral interventions for new settings and target populations. In: *AIDS Education & Prevention* 18 (2006), Nr. supp, S. 59–73
- [100] DUNKELS, Adam: The contikimac radio duty cycling protocol. In: *Forschungsbericht T2011:13, Swedish Institute of Computer Science*, 2011
- [101] BUETTNER, Michael ; YEE, Gary V. ; ANDERSON, Eric ; HAN, Richard: X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In: *Proceedings of the 4th international conference on Embedded networked sensor systems*. Boulder, USA, 2006, S. 307–320
- [102] ZHENG, Tao ; RADHAKRISHNAN, Sridhar ; SARANGAN, Venkatesh: PMAC: an adaptive energy-efficient MAC protocol for wireless sensor networks. In: *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium*. Denver, USA, 2005, S. 65–73
- [103] RAJENDRAN, Venkatesh ; OBRACZKA, Katia ; GARCIA-LUNA-ACEVES, J. J.: Energy-efficient Collision-free Medium Access Control for Wireless Sensor Networks. In: *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*. Los Angeles, USA, 2003, S. 181–192
- [104] SINGH, Shio K. ; SINGH, MP ; SINGH, DK u. a.: Routing protocols in wireless sensor networks—A survey. In: *International Journal of Computer Science & Engineering Survey* 1 (2010), Nr. 2, S. 63–83
- [105] XU, Ya ; HEIDEMANN, John ; ESTRIN, Deborah: Geography-informed energy conservation for ad hoc routing. In: *Proceedings of the 7th annual international conference on Mobile computing and networking*. Rome, Italien, 2001, S. 70–84

- [106] YU, Yan ; GOVINDAN, Ramesh ; ESTRIN, Deborah: Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. In: *Forschungsbericht ucla/csd-tr-01-0023, UCLA Computer Science Department*, 2001
- [107] INTANAGONWIWAT, Chalermek ; GOVINDAN, Ramesh ; ESTRIN, Deborah: Directed diffusion: a scalable and robust communication paradigm for sensor networks. In: *Proceedings of the 6th annual international conference on Mobile computing and networking*. Boston, Massachusetts, USA, 2000, S. 56–67
- [108] BRAGINSKY, David ; ESTRIN, Deborah: Rumor routing algorithm for sensor networks. In: *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. Atlanta, USA, 2002, S. 22–31
- [109] HEINZELMAN, Wendi R. ; CHANDRAKASAN, Anantha ; BALAKRISHNAN, Hari: Energy-efficient communication protocol for wireless microsensor networks. In: *Proceedings of the 33rd Hawaii International Conference on System Sciences*. Maui, USA, 2000, S. 8020–8030
- [110] MANJESHWAR, Arati ; AGRAWAL, Dharma P.: APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks. In: *Proceedings of the 16th International Parallel and Distributed Processing Symposium*. Fort Lauderdale, USA, 2002, S. 48–56
- [111] LUO, Jun ; PANCHARD, Jacques ; PIÓRKOWSKI, Michał ; GROSSGLAUSER, Matthias ; HUBAUX, Jean-Pierre: Mobiroute: Routing towards a mobile sink for improving lifetime in sensor networks. In: *Proceedings of the IEEE International Conference on Distributed Computing in Sensor Networks*. San Francisco, USA, 2006, S. 480–497
- [112] SHAH, Rahul C. ; ROY, Sumit ; JAIN, Sushant ; BRUNETTE, Waylon: Data MULEs: modeling a three-tier architecture for sparse sensor networks. In: *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*. Anchorage, USA, 2003, S. 30–41
- [113] LINDSEY, Stephanie ; RAGHAVENDRA, Cauligi ; SIVALINGAM, Krishna M.: Data gathering algorithms in sensor networks using energy metrics. In: *IEEE Transactions on Parallel and Distributed Systems* 13 (2002), Nr. 9, S. 924–935
- [114] DU, Xiaojiang ; LIN, Fengjing: Improving routing in sensor networks with heterogeneous sensor nodes. In: *Proceedings of the 61st IEEE Vehicular Technology Conference* Bd. 4 IEEE, 2005, S. 2528–2532



- [115] HE, Tian ; STANKOVIC, John A. ; LU, Chenyang ; ABDELZAHER, Tarek: SPEED: A stateless protocol for real-time communication in sensor networks. In: *Proceedings of the 23rd International Conference on Distributed Computing Systems*. Providence, USA, 2003, S. 46–55
- [116] COBO, Luis ; QUINTERO, Alejandro ; PIERRE, Samuel: Ant-based routing for wireless multimedia sensor networks using multiple QoS metrics. In: *Computer networks* 54 (2010), Nr. 17, S. 2991–3010
- [117] SHARP: *Datenblatt GP2Y0A02YK Long Distance Measuring Sensor*. <http://www.sharp-world.com>, [online] letzter Aufruf: 28.02.2016
- [118] HANWAI ELECTRONICS: *Datenblatt MD62 Thermal Conductor CO2 Gas Sensor*. <http://www.hwsensor.com>, [online] letzter Aufruf: 28.02.2016
- [119] OSRAM OPTO SEMICONDUCTORS: *Datenblatt SFH5711 High Accuracy Ambient Light Sensor*. <http://www.osram-os.com>, [online] letzter Aufruf: 28.02.2016
- [120] STMICROELECTRONICS: *Datenblatt LIS3MDL Digital output magnetic sensor: ultra low-power, high performance 3-axis magnetometer*. <http://www.st.com>, [online] letzter Aufruf: 28.02.2016
- [121] NAVILOCK: *Datenblatt NL-501ETTL Sirf3 TTL Modul*. <http://www.navilock.com>, [online] letzter Aufruf: 28.02.2016
- [122] VTI TECHNOLOGIES OY: *Datenblatt CMA3000-D01 3-AXIS ULTRA LOW POWER ACCELEROMETER WITH DIGITAL SPI AND I2C INTERFACE*. <http://www.vti.fi>, [online] letzter Aufruf: 28.02.2016
- [123] BOSCH SENSORTREC: *Datenblatt BMP085 Digital Pressure Sensor*. <http://www.bosch-sensortec.com>, [online] letzter Aufruf: 28.02.2016
- [124] B.I.O-TECH E.K: *Datenblatt FCH-m-PP-LC Mini Durchflussmesser*. <http://www.btflowmeter.com>, [online] letzter Aufruf: 28.02.2016
- [125] SENSIRION: *Datenblatt SHT21 Humidity and Temperature Sensor*. <http://www.sensirion.com>, [online] letzter Aufruf: 28.02.2016
- [126] HYGROSENS INSTRUMENTS GMBH: *Datenblatt Bewegungsmelder PIR-STD-LP*. <http://www.hygrosens.com>, [online] letzter Aufruf: 28.02.2016
- [127] RAGHUNATHAN, Vijay ; GANERIWAL, Saurabh ; SRIVASTAVA, Mani: Emerging techniques for long lived wireless sensor networks. In: *IEEE Communications Magazine* 44 (2006), Nr. 4, S. 108–114

- [128] ALIPPI, Cesare ; ANASTASI, Giuseppe ; DI FRANCESCO, Mario ; ROVERI, Manuel: An adaptive sampling algorithm for effective energy management in wireless sensor networks with energy-hungry sensors. In: *IEEE Transactions on Instrumentation and Measurement* 59 (2010), Nr. 2, S. 335–344
- [129] CHANG, Marcus ; BONNET, Philippe: Meeting ecologists’ requirements with adaptive data acquisition. In: *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. Zürich, Schweiz, 2010, S. 141–154
- [130] SHANNON, Claude E.: Communication in the presence of noise. In: *Proceedings of the IRE* 37 (1949), Nr. 1, S. 10–21
- [131] TSENG, Yu-Chee ; WANG, You-Cbiun ; CHENG, Kai-Yang ; HSIEH, Yao-Yu: iMouse: an integrated mobile surveillance and wireless sensor system. In: *IEEE Computer* 40 (2007), Nr. 6, S. 60–66
- [132] SINGH, Amarjeet ; BUDZIK, Diane ; CHEN, Willie ; BATALIN, Maxim A. ; STEALEY, Michael ; BORGSTROM, Henrik ; KAISER, William J.: Multiscale sensing: a new paradigm for actuated sensing of high frequency dynamic phenomena. In: *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Peking, China, 2006, S. 328–335
- [133] PADHY, Paritosh ; DASH, Rajdeep K. ; MARTINEZ, Kirk ; JENNINGS, Nicholas R.: A utility-based sensing and communication model for a glacial sensor network. In: *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. Hakodate, Japan, 2006, S. 1353–1360
- [134] DESHPANDE, Amol ; GUESTRIN, Carlos ; MADDEN, Samuel R. ; HELLERSTEIN, Joseph M. ; HONG, Wei: Model-driven data acquisition in sensor networks. In: *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. Toronto, Kanada, 2004, S. 588–599
- [135] ANASTASI, Giuseppe ; CONTI, Marco ; DI FRANCESCO, Mario ; PASSARELLA, Andrea: Energy conservation in wireless sensor networks: A survey. In: *Ad Hoc Networks* 7 (2009), Nr. 3, S. 537–568
- [136] LI, Ming ; GANESAN, Deepak ; SHENOY, Prashant: Presto: feedback-driven data management in sensor networks. In: *IEEE/ACM Transactions on Networking* 17 (2009), Nr. 4, S. 1256–1269
- [137] SANTINI, Silvia ; RÖMER, Kay: An adaptive strategy for quality-based data reduction in wireless sensor networks. In: *Proceedings of the 3rd International Conference on Networked Sensing Systems*. Chicago, USA, 2006, S. 29–36

- [138] LANDSIEDEL, Olaf ; WEHRLE, Klaus ; GÖTZ, Stefan: Accurate prediction of power consumption in sensor networks. In: *Proceedings of the Second Workshop on Embedded Networked Sensors*. Sydney, Australia, 2005, S. 37–44
- [139] CROSSBOW TECHNOLOGY: *Datenblatt Mica2*. <http://www.xbow.com>, [online] letzter Aufruf: 15.04.2013
- [140] GARG, Siddharth ; MARCULESCU, Diana: On the impact of manufacturing process variations on the lifetime of sensor networks. In: *Proceedings of the 5th IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*. Salzburg, Österreich, 2007, S. 203–208
- [141] BORKAR, Shekhar ; KARNIK, Tanay ; NARENDRA, Siva ; TSCHANZ, Jim ; KESHAVARZI, Ali ; DE, Vivek: Parameter variations and impact on circuits and microarchitecture. In: *Proceedings of the 40th annual Design Automation Conference*. Anaheim, USA, 2003, S. 338–342
- [142] WANNER, Lucas ; APTE, Charwak ; BALANI, Rahul ; GUPTA, Puneet ; SRIVASTAVA, Mani: A case for opportunistic embedded sensing in presence of hardware power variability. In: *Proceedings of the 2010 international conference on Power aware computing and systems*. Vancouver, Kanada, 2010, S. 1–8
- [143] ATMEL: *Datenblatt ATSAM3U4E AT91SAM ARM-based Flash MCU SAM3U Series*. <http://atmel.com>, [online] letzter Aufruf: 29.02.2016
- [144] TEXAS INSTRUMENTS: *Datenblatt eZ430-Chronos Development Tool*. <http://www.ti.com>, [online] letzter Aufruf: 27.02.2016
- [145] HITEX DEVELOPMENT TOOLS GMBH: *Datenblatt PowerScale*. <http://www.hitex.com>, [online] letzter Aufruf: 29.02.2016
- [146] HAMEG INSTRUMENTS GMBH: *Datenblatt HM8143*. <http://www.hameg.com>, [online] letzter Aufruf: 29.02.2016
- [147] XUE, Chun J. ; ZHANG, Youtao ; CHEN, Yiran ; SUN, Guangyu ; YANG, J. J. ; LI, Hai: Emerging Non-volatile Memories: Opportunities and Challenges. In: *Proceedings of the Seventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*. Taipei, Taiwan, 2011, 325–334
- [148] SUZUKI, Kosuke ; SWANSON, Steven: A Survey of Trends in Non-Volatile Memory Technologies: 2000-2014. In: *Proceedings of the Seventh IEEE International Memory Workshop*. Paris, Frankreich, 2015, S. 1–4

- [149] TEXAS INSTRUMENTS: *Datenblatt TPL5010 Nano-power SystemTimer with Watchdog Function*. <http://www.ti.com>, [online] letzter Aufruf: 27.02.2016
- [150] POP, Valer ; BERGVELD, Henk J. ; NOTTEN, PHL ; REGTIEN, Paul P.: State-of-the-art of battery state-of-charge determination. In: *Measurement Science and Technology* 16 (2005), Nr. 12, S. R93
- [151] MAXIM INTEGRATED: *Datenblatt DS2438 Smart Battery Monitor*. <http://www.maximintegrated.com>, [online] letzter Aufruf: 29.02.2016
- [152] TEXAS INSTRUMENTS: *Datenblatt BQ26500 Single Cell Li-Ion and Li-Pol Battery Gas Gauge*. <http://www.ti.com>, [online] letzter Aufruf: 29.02.2016
- [153] MAXIM INTEGRATED: *Datenblatt MAX17043/MAX17044 Compact, Low-Cost 1S/2S Fuel Gauges with Low-Battery Alert*. <http://www.maximintegrated.com>, [online] letzter Aufruf: 29.02.2016
- [154] UNIONFORTUNE ELECTRONIC CO, LTD: *Datenblatt 110mAh polymer lithium-ion battery*. <http://www.unionfortune.com>, [online] letzter Aufruf: 29.02.2016
- [155] DOYLE, Marc ; FULLER, Thomas F. ; NEWMAN, John: Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. In: *Journal of the Electrochemical Society* 140 (1993), Nr. 6, S. 1526–1533
- [156] NEWMAN, John: *Fortran programs for simulation of electrochemical systems, dualfoil.f program for lithium battery simulation*. <http://www.cchem.berkeley.edu/jsngrp/fortran.html>, [online] letzter Aufruf: 29.02.2016
- [157] GOLD, Sean: A PSPICE macromodel for lithium-ion batteries. In: *Proceedings of the 12th Annual Battery Conference on Applications and Advances*. Long Beach, USA, 1997, S. 215–222
- [158] BENINI, Luca ; CASTELLI, Giuliano ; MACII, Alberto ; MACII, Enrico ; PONCINO, Massimo ; SCARSI, Riccardo: Discrete-time battery models for system-level low-power design. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 9 (2001), Nr. 5, S. 630–640
- [159] VASEBI, Amir ; PARTOVIBAKHSH, Maral ; BATHAEE, S: A novel combined battery model for state-of-charge estimation in lead-acid batteries based on extended Kalman filter for hybrid electric vehicle applications. In: *Journal of Power Sources* 174 (2007), Nr. 1, S. 30–40
- [160] CHIASSERINI, C-F ; RAO, Ramesh R.: Energy efficient battery management. In: *IEEE Journal on Selected Areas in Communications* 19 (2001), Nr. 7, S. 1235–1245

- [161] PANIGRAHI, T ; PANIGRAHI, D ; CHIASSERINI, C ; DEY, S ; RAO, R ; RAGHUNATHAN, Anand ; LAHIRI, K u. a.: Battery life estimation of mobile embedded systems. In: *Proceedings of the The 14th International Conference on VLSI Design*. Bangalore, Indien, 2001, S. 57–63
- [162] CAI, Chenghui ; DU, Dong ; LIU, Zhiyu ; GE, Jingtian: State-of-charge (SOC) estimation of high power NI-MH rechargeable battery with artificial neural network. In: *Proceedings of the 9th International Conference on Neural Information Processing* Bd. 2. Singapur, 2002, S. 824–828
- [163] CHANG, Wen-Yeau: State of Charge Estimation for LiFePO<sub>4</sub> Battery Using Artificial Neural Network. In: *International Review of Electrical Engineering* 7 (2012), Nr. 5, S. 5874–5880
- [164] SALKIND, Alvin J. ; FENNIE, Craig ; SINGH, Pritpal ; ATWATER, Terrill ; REISNER, David E.: Determination of state-of-charge and state-of-health of batteries by fuzzy logic methodology. In: *Journal of Power Sources* 80 (1999), Nr. 1, S. 293–300
- [165] SINGH, Pritpal ; VINJAMURI, Ramana ; WANG, Xiquan ; REISNER, David: Design and implementation of a fuzzy logic-based state-of-charge meter for Li-ion batteries used in portable defibrillators. In: *Journal of power sources* 162 (2006), Nr. 2, S. 829–836
- [166] RAKHMATOV, Daler N. ; VRUDHULA, Sarma B.: An analytical high-level battery model for use in energy management of portable electronic systems. In: *Proceedings of the 2001 IEEE/ACM international conference on Computer-aided design*. San Jose, USA, 2001, S. 488–493
- [167] BUCHLI, Bernhard ; ASCHWANDEN, Daniel ; BEUTEL, Jan: Battery state-of-charge approximation for energy harvesting embedded systems. In: *Proceedings of the 10th European Conference on Wireless Sensor Networks*. Ghent, Belgien, 2013, S. 179–196
- [168] LACHENMANN, Andreas: *A cross-layer framework for sensor networks*. Stuttgart, Universität Stuttgart, Dissertation, 2008
- [169] CASTAGNETTI, Andrea ; PEGATOQUET, Alain ; BELLEUDY, Cécile ; AUGUIN, Michel: An efficient state of charge prediction model for solar harvesting wsn platforms. In: *Proceedings of the 19th International Conference on Systems, Signals and Image Processing*. Wien, Österreich, 2012, S. 122–125
- [170] KELLNER, Simon ; BELLOSA, Frank: Energy accounting support in tinyos. In: *PIK-Praxis der Informationsverarbeitung und Kommunikation* 32 (2009), Nr. 2, S. 105–109

- [171] DUNKELS, Adam ; OSTERLIND, Fredrik ; TSIFTES, Nicolas ; HE, Zhitao: Software-based on-line energy estimation for sensor nodes. In: *Proceedings of the 4th workshop on Embedded networked sensors*. Cork, Irland, 2007, S. 28–32
- [172] SIEBER, André ; NOLTE, Jörg: Online device-level energy accounting for wireless sensor nodes. In: *Proceedings of the 10th European conference on Wireless Sensor Networks*. Ghent, Belgien, 2013, S. 149–164
- [173] MENZEL, Thomas ; WOLISZ, Adam: Overpotential-based Battery End-of-Life Indication in WSN Nodes. In: *Proceedings of The 4th International Conference on Sensor Systems and Software*, 2013, S. 34–46
- [174] PANASONIC: *Datenblatt eneloop BK-4LCC AA Nickel-Metal Hydride Battery*. <http://www.eneloop.eu/>, [online] letzter Aufruf: 27.02.2016
- [175] PROCTER & GAMBLE: *Datenblatt Duracell CR2032 Lithium Manganese Dioxide Battery*. <http://www.duracell.com>, [online] letzter Aufruf: 27.02.2016
- [176] SHNAYDER, Victor ; HEMPSTEAD, Mark ; CHEN, Bor-rong ; ALLEN, Geoff W. ; WELSH, Matt: Simulating the power consumption of large-scale sensor network applications. In: *Proceedings of the 2nd international conference on Embedded networked sensor systems*. Baltimore, USA, 2004, S. 188–200
- [177] VARGA, András: The OMNeT++ Discrete Event Simulation System. In: *Proceedings of the European Simulation Multiconference*. Prag, Tschechische Republik, 2001, S. 185
- [178] LANDSIEDEL, Olaf ; WEHRLE, Klaus ; GÖTZ, Stefan: Accurate prediction of power consumption in sensor networks. In: *Proceedings of the 2nd Workshop on Embedded Networked Sensors*. Sydney, Australien, 2005, S. 37–44
- [179] JIANG, Xiaofan ; DUTTA, Prabal ; CULLER, David ; STOICA, Ion: Micro power meter for energy monitoring of wireless sensor networks at scale. In: *Proceedings of the 6th international conference on Information processing in sensor networks*. Cambridge, USA, 2007, S. 186–195
- [180] DUTTA, Prabal ; FELDMEIER, Mark ; PARADISO, Joseph ; CULLER, David: Energy metering for free: Augmenting switching regulators for real-time monitoring. In: *Proceedings of the 7th International Conference on Information Processing in Sensor Networks*. St. Louis, USA, 2008, S. 283–294



- [181] DONG, Mian ; ZHONG, Lin: Self-constructive high-rate system energy modeling for battery-powered mobile systems. In: *Proceedings of the 9th international conference on Mobile systems, applications, and services*. Bethesda, USA, 2011, S. 335–348
- [182] HERGENRÖDER, Anton ; HORNEBER, Jens ; MEIER, Detlev ; ARMBRUSTER, Patrick ; ZITTERBART, Martina: Distributed energy measurements in wireless sensor networks. In: *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. Berkeley, USA, 2009, S. 299–300
- [183] LACHENMANN, Andreas ; MARRÓN, Pedro J. ; MINDER, Daniel ; ROTHERMEL, Kurt: Meeting lifetime goals with energy levels. In: *Proceedings of the 5th international conference on Embedded networked sensor systems*. Sydney, Australien, 2007, S. 131–144
- [184] SCHMIDT, Daniel ; KRÄMER, Marc ; KUHN, Thomas ; WEHN, Norbert: Energy modelling in sensor networks. In: *Advances in Radio Science* 5 (2007), S. 347–351
- [185] DUNKELS, Adam ; ERIKSSON, Joakim ; FINNE, Niclas ; TSIFTES, Nicolas: Powertrace: Network-level power profiling for low-power wireless networks. In: *Forschungsbericht T2011:05, Swedish Institute of Computer Science*, 2011
- [186] HURNI, Philipp ; NYFFENEGGER, Benjamin ; BRAUN, Torsten ; HERGENROEDER, Anton: On the accuracy of software-based energy estimation techniques. In: *Proceedings of the 8th European conference on Wireless Sensor Networks*. Bonn, Deutschland, 2011, S. 49–64
- [187] FONSECA, Rodrigo ; DUTTA, Prabal ; LEVIS, Philip ; STOICA, Ion: Quanto: tracking energy in networked embedded systems. In: *Proceedings of the 8th USENIX conference on Operating systems design and implementation*. San Diego, USA, 2008, S. 323–338
- [188] BELLOSA, Frank: The benefits of event: driven energy accounting in power-sensitive systems. In: *Proceedings of the 9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system*. Kolding, Dänemark, 2000, S. 37–42
- [189] PIOTROWSKI, K ; ORTMANN, S ; LANGENDÖRFER, P: Multi-radio wireless sensor node for mobile biomedical monitoring. In: *46. Jahrestagung der Deutschen Gesellschaft für Biomedizinische Technik*. Jena, Deutschland, 2012, S. 725–728

- [190] RENNER, Christian ; UNTERSCHÜTZ, Stefan ; TURAU, Volker: Power Management for Wireless Sensor Networks Based on Energy Budgets. In: *Hamburg University of Technology, Hamburg, Deutschland, Tech. Rep* (2011)
- [191] FLINN, Jason ; SATYANARAYANAN, M.: Energy-aware Adaptation for Mobile Applications. In: *Proceedings of the Seventeenth ACM Symposium on Operating Systems Principles*. Charleston, USA : ACM, 1999, 48–63
- [192] LIU, J. W S. ; SHIH, Wei-Kuan ; LIN, Kwei-Jay ; BETTATI, R. ; CHUNG, J.-Y.: Imprecise computations. In: *Proceedings of the IEEE* 82 (1994), Nr. 1, S. 83–94
- [193] CHALLEN, Geoffrey W. ; WATERMAN, Jason ; WELSH, Matt: IDEA: Integrated Distributed Energy Awareness for Wireless Sensor Networks. In: *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*. San Francisco, USA, 2010, S. 35–48
- [194] JIANG, Xiaofan ; TANEJA, Jay ; ORTIZ, Jorge ; TAVAKOLI, Arsalan ; DUTTA, Prabal ; JEONG, Jaemin ; CULLER, David E. ; LEVIS, Philip ; SHENKER, Scott u.a.: An architecture for energy management in wireless sensor networks. In: *ACM SIGBED Review - Special issue on the workshop on wireless sensor network architecture* 4 (2007), Nr. 3, S. 31–36
- [195] WANNER, Lucas ; APTE, Charvak ; BALANI, Rahul ; GUPTA, Puneet ; SRIVASTAVA, Mani: Hardware variability-aware duty cycling for embedded sensors. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 21 (2013), Nr. 6, S. 1000–1012
- [196] KELLNER, Simon: Flexible online energy accounting in TinyOS. In: *Proceedings of the 4th International Workshop on Real-World Wireless Sensor Networks*. Colombo, Sri Lanka, 2010, S. 62–73
- [197] CAO, Qing ; FESEHAYE, Debessay ; PHAM, Nam ; SARWAR, Yusuf ; ABDELZAHER, Tarek: Virtual battery: An energy reserve abstraction for embedded sensor networks. In: *Proceedings of the 29th IEEE Real-Time Systems Symposium*. Barcelona, Spanien, 2008, S. 123–133
- [198] MAINLAND, Geoffrey ; PARKES, David C. ; WELSH, Matt: Decentralized, adaptive resource allocation for sensor networks. In: *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*. Berkeley, USA, 2005, S. 315–328



- [199] ROY, Arjun ; RUMBLE, Stephen M. ; STUTSMAN, Ryan ; LEVIS, Philip ; MAZIERES, David ; ZELDOVICH, Nikolai: Energy management in mobile devices with the Cinder operating system. In: *Proceedings of the sixth conference on Computer systems*. Salzburg, Österreich, 2011, S. 139–152
- [200] ZENG, Heng ; ELLIS, Carla S. ; LEBECK, Alvin R. ; VAHDAT, Amin: Currentcy: A Unifying Abstraction for Expressing Energy Management Policies. In: *Proceedings of the USENIX Annual Technical Conference*. San Antonio, USA, 2003, S. 43–56
- [201] LORINCZ, Konrad ; CHEN, Bor-rong ; WATERMAN, Jason ; WERNER-ALLEN, Geoff ; WELSH, Matt: Resource Aware Programming in the Pixie OS. In: *Proceedings of the 6th ACM conference on Embedded network sensor systems*. Raleigh, USA, 2008, S. 211–224
- [202] DIJKSTRA, Edsger W.: The mathematics behind the Banker’s algorithm. In: *Selected Writings on Computing: A Personal Perspective*. Springer, 1982, S. 308–312
- [203] VIRAG, Ana ; BOGDAN, Stjepan: Resource allocation in smart homes based on Banker’s algorithm. In: *Proceedings of the 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies*. Manchester, Vereinigtes Königreich, 2011, S. 1–7
- [204] LAGEMANN, Andreas ; NOLTE, Jörg: Integration of Event-Driven Embedded Operating Systems Into OMNet++ – A Case Study with Reflex. In: *Proceedings of the 2nd International Workshop on OMNeT++*. Rom, Italien, 2009, S. 73:1–73:7
- [205] GERWIN, Werner ; SCHAAF, Wolfgang ; BIEMELT, Detlef ; ELMER, Michael ; MAURER, Thomas ; SCHNEIDER, Anna: The Artificial catchment ‘Hühnerwasser’ (Chicken Creek): construction and initial properties. In: *Ecosystem Development*, 2010. – ISSN 1867–7800, S. 1–56
- [206] MENZEL, Hannes ; KARNAPKE, Reinhardt ; BUCHNER, Steffan ; NOLTE, Jörg: Applying wireless sensor networks to renaturation monitoring: From requirements to deployment. In: *Proceedings of Wireless Days 2014*. Rio de Janeiro, Brasilien, 2014, S. 1–6
- [207] MAURER, Thomas. ; SCHAPP, Andrea ; BÜCHNER, Steffen ; MENZEL, Hannes ; HINZ, C.: Measurement of rainfall distribution on a small catchment for the evaluation of canopy interception effects. In: *EGU General Assembly Conference Abstracts* Bd. 16, 2014 (EGU General Assembly Conference Abstracts), S. 12176

- [208] SENSIRION: *Datenblatt SHT11 Humidity and Temperature Sensor*. <http://www.sensirion.com>, [online] letzter Aufruf: 28.02.2016